(12)

RADC-TR-77-17
Technical Report
January 1977

IMAGE UNDERSTANDING AND INFORMATION EXTRACTION

Purdue University

Approved for public release; distribution unlimited.

Sponsored by
Defense Advanced Research Projects Agency (DoD)
ARPA Order No. 2893

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U. S. Government.

ROME AIR DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
GRIFFISS AIR FORCE BASE, NEW YORK 13441

This report has been reviewed by the RADC Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

This report has been reviewed and approved for publication.

APPROVED: David J. Brazil

DAVID J. BRAZIL, Capt, USAF
Project Engineer

Do not return this copy. Retain or destroy.

IMAGE UNDERSTANDING AND INFORMATION EXTRACTION

T. S. Huang
K. S. Fu

Contractor: Purdue University
Contract Number: F30602-75-C-0150
Effective Date of Contract: 1 November 1975
Contract Expiration Date: 31 October 1976
Short Title of Work: Image Understanding and Information Extraction
Program Code Number: 5D30
Period of Work Covered: May - Jul 76

Principal Investigators: Dr. Thomas S. Huang and Dr. King Sun Fu
Phone: 317 493-3361
Project Engineer: Capt David J. Brazil
Phone: 315 330-3175

This research was supported by the Defense Advanced Research Projects Agency of the Department of Defense and was monitored by Capt David J. Brazil (IRRE), Griffiss AFB NY 13441 under Contract F30602-75-C-0150.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 18 19 RADC-TR-77-17	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) 6 IMAGE UNDERSTANDING AND INFORMATION EXTRACTION		5. TYPE [redacted] 9 ~~Third~~ Quarterly Rep[redacted]t. no. 3, 1 May - 31 July [redacted]76.
		6. PERFORMING ORG. REPORT NUMBER N/A
7. AUTHOR(s) 10 T. S. Huang K. S. Fu		8. CONTRACT OR GRANT NUMBER(s) 15 F30602-75-C-0150, ARPA Order-2893
9. PERFORMING ORGANIZATION NAME AND ADDRESS Purdue University/Department of Electrical Engineering W. Lafayeete IN 47907		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61101E B893001
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Advanced Research Projects Agency 1400 Wilson Blvd Arlington VA 22209		12. REPORT DATE 11 Jan[redacted]77
		13. NUMBER OF PAGES 140 12 146p.
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Rome Air Development Center (IRRE) Griffiss AFB NY 13441		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (of this Report)

16 B893

Approved for public release; distribution unlimited.

17 00

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer:
Capt David J. Brazil (IRRE)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Digital image processing; image segmentation; image texture measurement; syntactic pattern recognition; Fourier descriptors.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report summarizes the results of our research program on Image Understanding and Information Extraction supported by the Defense Advanced Research Projects Agency under Contract F30602-75-C-0150. The report covers the period 1 May to 31 July 1976.

The objective of our research is to achieve a better understanding of image structure and to use this knowledge to develop techniques for image analysis

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

408124 JB

and processing tasks, especially information extraction. Our emphasis is on syntactic decomposition and recognition of imagery based on scene analysis. It is our hope that the results of this research will form the basis for the development of technology relevant to military applications of machine extraction of information from aircraft and satellite imagery.

TABLE OF CONTENTS

IMAGE UNDERSTANDING AND INFORMATION EXTRACTION

Research Summary

This report summarizes our research progress for the period May 1, 1976 through July 31, 1976 in Image Understanding and Information Extraction. The objective of this research is to achieve better understanding of image structure and to improve the capability of image data processing systems for extracting information from imagery and conveying that information in a useful form. The results of this research are expected to form the basis for technology relevant to military applications of machine extraction of information from aircraft and satellite imagery.

Our research projects can be categorized into six heavily overlapping areas: Image Segmentation, Image Attributes, Image Structure, Image Recognition Techniques, Preprocessing, and Applications.

IMAGE SEGMENTATION - In the previous quarterly report, we described a technique for accurately estimating edge locations which is useful for applications requiring mensuration. Burnett and Huang have continued to pursue this work, which uses a discrete position finite-state Markov process model to produce accurate width estimates from blurred and nonlinear observations in the presence of signal-dependent noise. It is shown here that the proposed algorithm is optimal when the states are known *a priori*. Experimental results are given for the case where the states are estimated from the available data.

Taking a different tack, Yoo and Huang report a clustering approach to image segmentation. This approach, somewhat different from earlier approaches to segmentation by clustering, involves four relatively distinct steps: (1) feature extraction, (2) clustering of the features in the feature space, (3) transformation of the clustering results back into the image, and (4) segmentation based on cluster boundaries in the image. Examples of applying this approach to various images are provided.

IMAGE ATTRIBUTES - We continue to pursue the analysis of shape and texture in images. Results of recent progress with Fourier shape descriptors appear in the APPLICATIONS section. Some new results of our texture research are described by Carlton and Mitchell.

IMAGE STRUCTURE - Tree grammars have proved to be a useful approach for characterizing the syntax or structure of images. In an extensive report Fu and Keng describe the use of tree-grammatical rules for the description of "objects" such as highways and rivers. By using additional semantic information, they have extended their method to the problem of recognizing bridges.

Further results of using a syntactic approach appear in the APPLICATIONS section.

IMAGE RECOGNITION TECHNIQUES - Pursuing the use of contextual information for statistical classification, Fu et al. have discovered that the form of the joint probability measure defined for the "random field" description of the image must meet certain functional constraints. This may not prove to be a serious restriction; however, further results are not yet available. They are also developing simulated data sets which will help to evaluate methods proposed for extracting spatial (2-dimensional) infomration from multispectral remote sensing data.

PREPROCESSING - Two-dimensional complex cepstrum analysis has been shown previously to be a means for stability analysis of two-dimensional recursive filters. O'Connor and Huang discuss the use of this form of analysis for enhancement of images blurred by certain point-spread functions. They are developing a software realization using the Fast Fourier Transform and have applied it successfully to filter stability analysis.

Berger and Huang have experimentally compared two methods for image restoration in the presence of noise. The Projection Method and the Singular

Value Decomposition did not yield very different results in the cases investigated, over a significant range of noise levels. From a practical standpoint, however, they found the Projection Method can utilize a priori information about the image, if available, and is more efficiently applied to images of larger size.

APPLICATIONS - Proceeding with their work, reported earlier, using Fourier shape descriptors for the analysis of airplane shapes, Wallace and Wintz have developed a normalization method which is not susceptible to noise problems as have been previously reported methods. They are now anticipating the integration of their method with automatic boundary-finding procedures in order to automatically detect and recognize airplanes.

Dang and Huang report some preliminary results from their work in locating airports in LANDSAT imagery. They are using a combination of spatial frequency filtering and syntactic analysis.

Mitchell describes the directions being pursued in the recognition of tactical targets in FLIR imagery. This is a joint project with Honeywell Systems and Research Division.

IMAGE MEASUREMENT

J.W. Burnett and T.S. Huang

I. Introduction

Our last report [1] showed how a discrete position finite state Markov-process model could be used in conjunction with the Viterbi algorithm (VA) to produce accurate width estimates from blurred and nonlinear observations in the presence of signal dependent noise. This report shows the algorithm is optimal when the states are known a priori, and presents experimental results on the algorithm's performance when the states are estimated from the available data.

The Optimality of the VA for Discrete Step Edge Location

Recall [1] that a sequence $\underline{\xi}$ will be decided over any other sequence $\hat{\underline{\xi}}$ when

$$p(\underline{Z}|\underline{\xi})\ P(\underline{\xi}) > p(\underline{Z}|\hat{\underline{\xi}})\ P(\hat{\underline{\xi}}) \tag{1}$$

As before let all permissible sequences be equally likely. Then (1) becomes (for all permissible $\underline{\xi}$ and $\hat{\underline{\xi}}$) decide ξ over $\hat{\underline{\xi}}$ if

$$p(\underline{Z}|\xi) > p(\underline{Z}|\hat{\xi}) \tag{2}$$

It is well known [2] that this decision rule minimizes the probability of deciding sequence $\hat{\xi}$ when $\underline{\xi}$ is the correct sequence. Now for step edges, each $\hat{\underline{\xi}}$ uniquely corresponds to an edge location. Thus the estimate of a step edge location produced by the VA (with the stated assumptions) is the minimum probability of error estimate. Therefore if π_α^* is the probability that the edge is mislocated α points by the VA and π_α is the probability the edge is mislocated α points by any other technique $\pi_\alpha^* \leq \pi_\alpha$. Let n^* be the number of points the edge is mislocated by the VA. Let n ($n=\pm 1, \pm 2, \ldots$) be the number of points the edge is mislocated by any other technique. Then

$$|\overline{n}| - |\overline{n}^*| = E|n| - E|n^*| = \sum_{\alpha} |\alpha| \pi_\alpha - \sum_{\alpha} |\alpha| \pi^*_\alpha$$

$$= \sum_{\alpha} |\alpha| (\pi_\alpha - \pi^*_\alpha)$$

$$\geq 0$$

$$\text{or } E|n| \geq E|n^*| \tag{3}$$

Equation (3) shows that on the average the accuracy of the VA cannot be improved upon. Further if $|\overline{n}| = |\overline{n}^*|$ then

$$\text{Var}(|n|) - \text{Var}(|n^*|) = \sum_{\alpha} (|\alpha| - |\overline{n}|)^2 \pi_\alpha - \sum_{\alpha} (|\alpha| - |\overline{n}^*|)^2 \pi_\alpha$$

$$= \sum_{\alpha} \alpha^2 (\pi_\alpha - \pi^*_\alpha) \geq 0 \tag{4}$$

Thus the variance of the edge location estimates produced by the VA cannot be bettered by any technique that is as accurate.

The Optimality of the VA for Discrete Width Measurement

With the assumption of independent edges let

$$p^* = Pr(n^*_W = \alpha) = \sum_{\beta} \pi^*_\beta \pi^*_{\alpha\ \beta} \tag{5a}$$

and

$$p_\alpha = Pr(n_W = \alpha) = \sum_{\beta} \pi_\beta \pi^*_{\alpha-\beta} \tag{5b}$$

As noted in the previous section $\pi_\gamma \leq \pi_\gamma$ for any γ. Thus term by term the summation in (5a) is less than or equal to the summation in (5b). Therefore,

$$p^*_\alpha \leq p_\alpha. \tag{6}$$

By arguments similar to those in the preceding section

$$|n_w^*| \leq E|n_w| \tag{7}$$

and

$$\mathrm{Var}\ (|n_w^*|) \leq \mathrm{Var}\ (|n_w|) \tag{8}$$

providing $E|n_w^*| = E|n_w|$. These last two equations show that the VA produces the most accurate and minimum variance discrete width estimates possible.

<u>Relation to the Continuous Case</u>

A single measurement by the VA will be an integer number of sample points. Since there is no physical reason the size of any given object will be an integer multiple of the sampling interval some constraint must be placed on the interval between points to prevent loss of measurement accuracy due to the discreteness of the estimates.

Suppose the width of an object is considered to be a continuous variable and that a maximum likelihood estimate of the continuous variable is available. If the signal to noise ratio is high enough the estimate $\hat{w}_C$ produced by this technique will be normally distributed with a mean value equal to the true width and a variance given by the Cramer-Rao lower bound on the variance of an unbiased estimate of w [3]. Since $\hat{w}_C$ is normally distributed with 95.4% confidence the true width can be expected to fall in the interval $[\hat{w}_C - 2\sigma_{CR}, \hat{w}_C + 2\sigma_{CR}]$ [3] where σ_{CR}^2 is the variance predicted by the Cramer-Rao lower bound. If for a particular problem $\hat{w}_C$ is 31.5 sample points and σ_{CR} is .1 sample points, it is reasonable to expect the true width is somewhere between 31.3 and 31.7 sample points. The closest width estimate the VA could produce would be 31 or possibly 32 sample points.

Assume for the moment the probability that the VA decided $\hat{w}_D$ ($\hat{w}_D$ an integer) sample points was the probability the continuous measurement fell in the interval $[\hat{w}_D, \hat{w}_D+1]$ sample points. The probability of deciding 32 sample points would be

$$Pr(\hat{w}_D=32) = Pr(\hat{w}_C > 32) = Pr(\hat{w}_C - 31.5 > .5)$$

$$= Pr(\frac{\hat{w}_C - 31.5}{.1} > 5) \approx 0$$

Similar arguments may be made to show $Pr(w_D = 29) \approx 0$. Thus with high probability repeated measurmenets will all have a value of 31 sample points so that unless there is "jitter" in the sample positions the bias will not be reduced by averaging several measurements.

The problem can be avoided by specifying that the interval ΔX between the samples is sufficiently small. As a rough guide

$$\Delta X \leq 2\sigma_{CR} \tag{9}$$

seems reasonable. This rule limits the bias due to the discreteness of a single measurement to a maximum of σ_{CR} and at least makes it possible for some reduction in bias to occur by averaging several measurements.

<u>Simulated Width Measurement</u>

A pulse with a width of thirty sample points simulating a scan line across an object to be measured was generated on a computer (see Fig. 1). The line was convolved with a Gaussian line spread function. The blur had a standard deviation of one sample point and was normalized so that the coefficients summed to unity. Each blurred intensity sample b_k was transformed to a density sample y_k by

$$y_k = \log b_k + .15 \tag{10}$$

to simulate the D - log E curve of film. Independent normally distributed

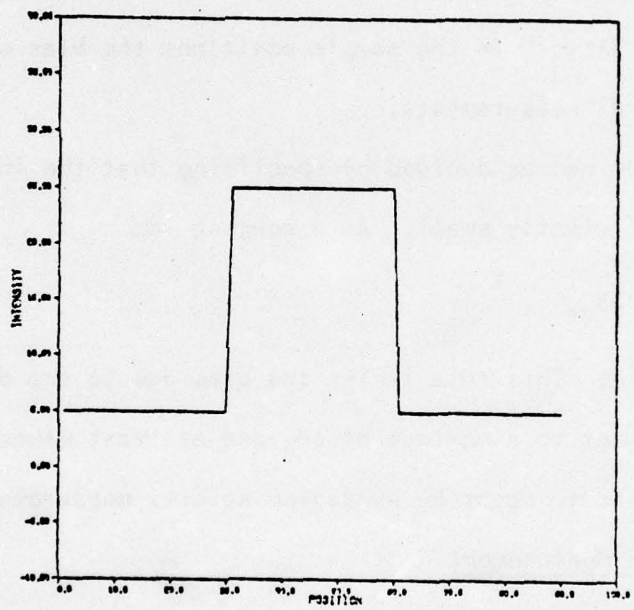


Figure 1 Ideal reflected light intensity

noise samples with zero mean and standard deviation

$$\sigma_{y_k} = .1\ y_k^{\frac{1}{3}} \tag{11}$$

were added to each y_k value.

Different SNR's were produced by varying the intensity levels of the original pulse. For this example the SNR was defined as

$$\mathrm{SNR} = \frac{|\text{maximum change in density due to signal}|}{\text{maximum noise standard deviation}} \tag{12}$$

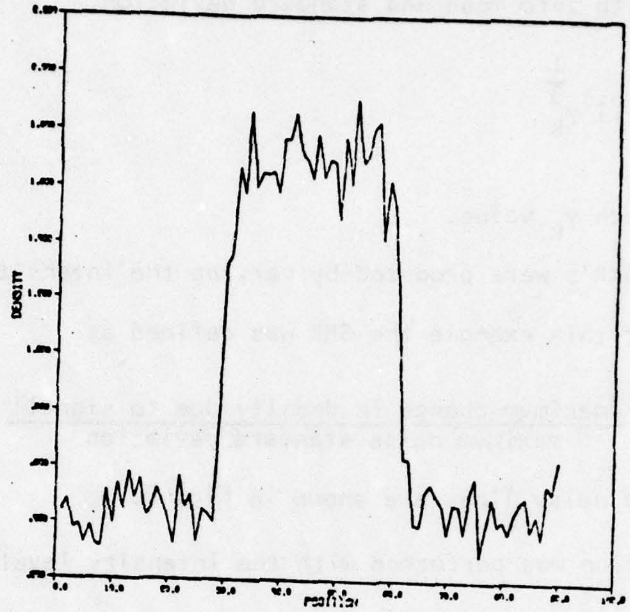
Some blurred and noisy lines are shown in Fig. 2.

The simulation was performed with the intensity levels assumed to be unknown. Eighteen training samples from each of the two density levels were selected. For high SNR's the selection of training samples that are well away from edge locations is easily accomplished by inspection of a single scan line. At low SNR's selection of training samples is easily done by inspection of the light intensity pattern for a SNR of 2.25 (see Fig. 3).

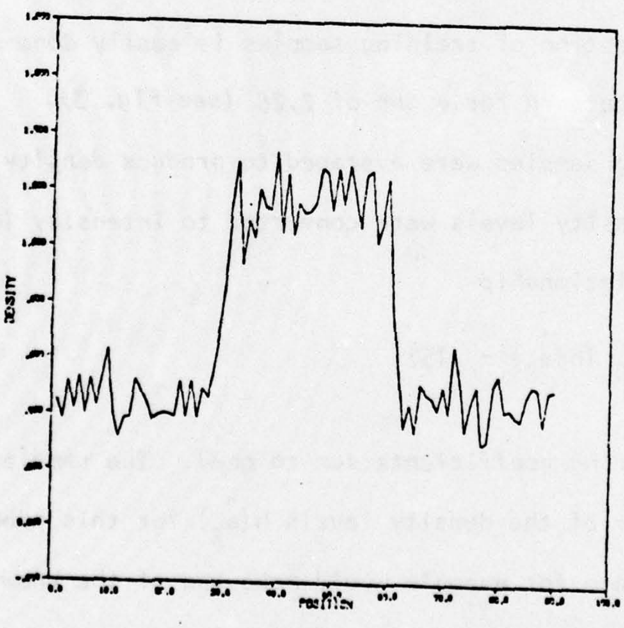
The training samples were averaged to produce density estimates $h(\hat{a}_1)$ and $h(\hat{a}_2)$. These density levels were converted to intensity levels $\hat{a}_1$ and $\hat{a}_2$ with the D - log E relationship

$$\hat{a}_\ell = 10^{(h(\hat{a}_\ell) - .15)}$$

(recall the blurring coefficients sum to one). The sample mean is not the optimal estimator of the density levels $h(\hat{a}_\ell)$ for this problem. The maximum likelihood estimate for example would make use of the knowledge that the variance of the samples also depends on the density levels. However, this estimate requires the roots of a polynomial be found. It was decided the additional computer time required to do root finding would probably not be worth the decreased varinace of the estimate and so the sample mean was chosen.

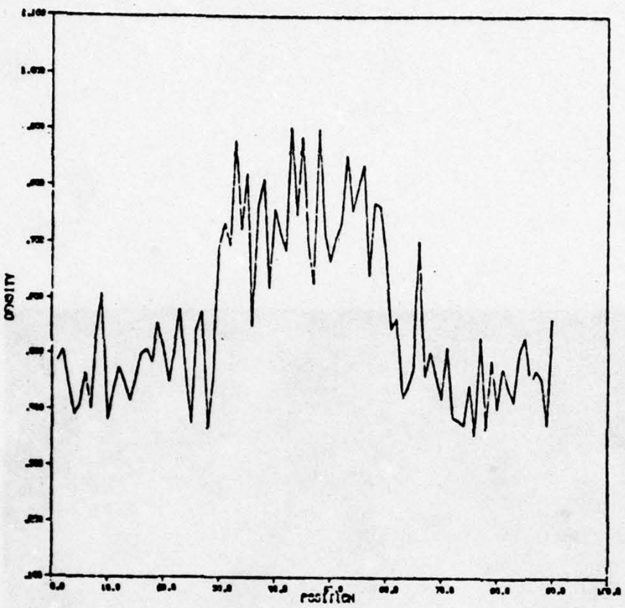


(a) SNR = 10.4

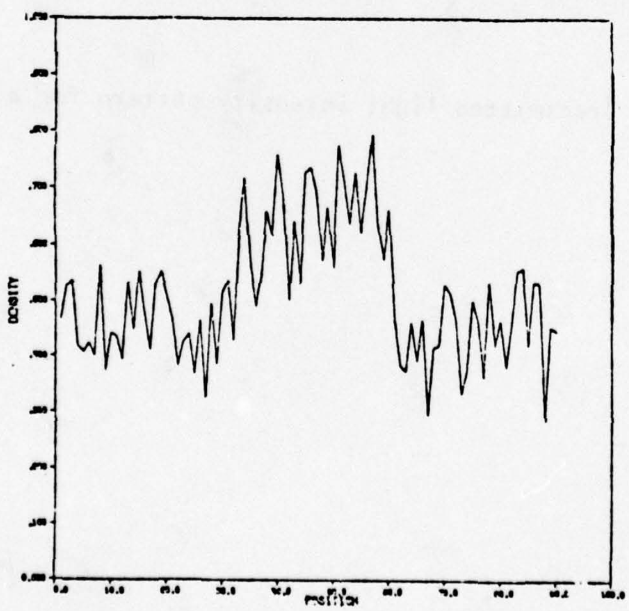


(b) SNR = 6.58

Figure 2 Typical blurred and noisy scan lines



(c) SNR = 3.3



(d) SNR = 2.25

Figure 2 cont.

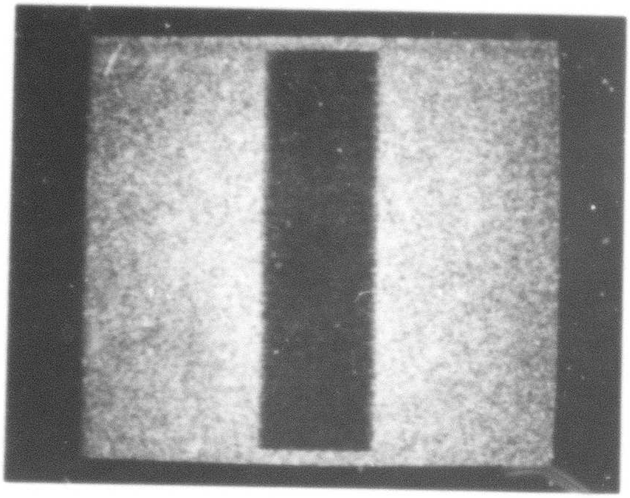


Figure 3 Transmitted light intensity pattern for a SNR of 2.25

The results are shown in Figures 4 and 5. Two hundred noise sample functions were used at each SNR. Figure 4 shows that estimating the levels from the available data has very little effect on accuracy. Figure 5 shows the variance of the estimate is increased slightly as expected from [1].

A Cylinder Problem

Consider a polished metallic cylinder on a black background of "infinite" extent. Illumination is from a light source "infinitely" far away so that all light rays striking the cylinder make a uniform angle ϕ with the horizontal (see Figure 6). The observer is located directly over the cylinder and "infinitely" far away.

The light intensity in the direction of a reflected ray is [6]

$$\text{reflected intensity} = CR_S(\alpha) \tag{13}$$

where C is the incident light intensity and $R_S(\alpha)$ is the ratio of reflected energy to incident energy as a function of the incident angle α. By the assumption on the observer's position and distance the light intensity in the direction of the observer is $CR_S(\alpha)\cos S$ where S is the angle between the reflected ray and the vertical. The problem is to measure the radius of the cylinder.

The problem of measuring the radius has several interesting aspects. First, the ideal reflected light intensity in the direction of the observer is not constant but varies with position on the cylinder. Further, surface areas of the cylinder that are not directly illuminated by the light source cannot be seen due to the assumption of a black background. Finally since no diffuse reflection occurs (due to the polished surface assumption) the areas of the cylinder that are directly illuminated but do not reflect incident light toward the observer also cannot be seen.

At any position X along a scan line across the cylinder the normal vector

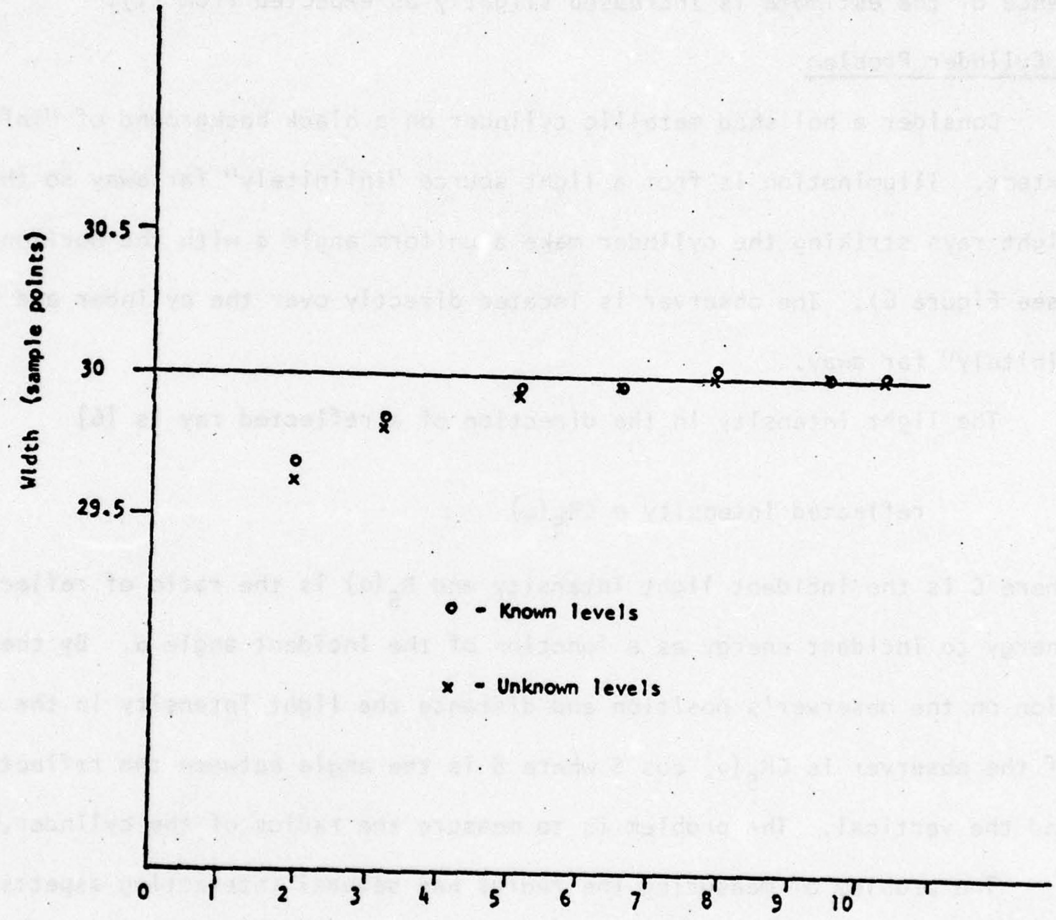


Figure 4 The effect of unknown levels on the accuracy of the estimates

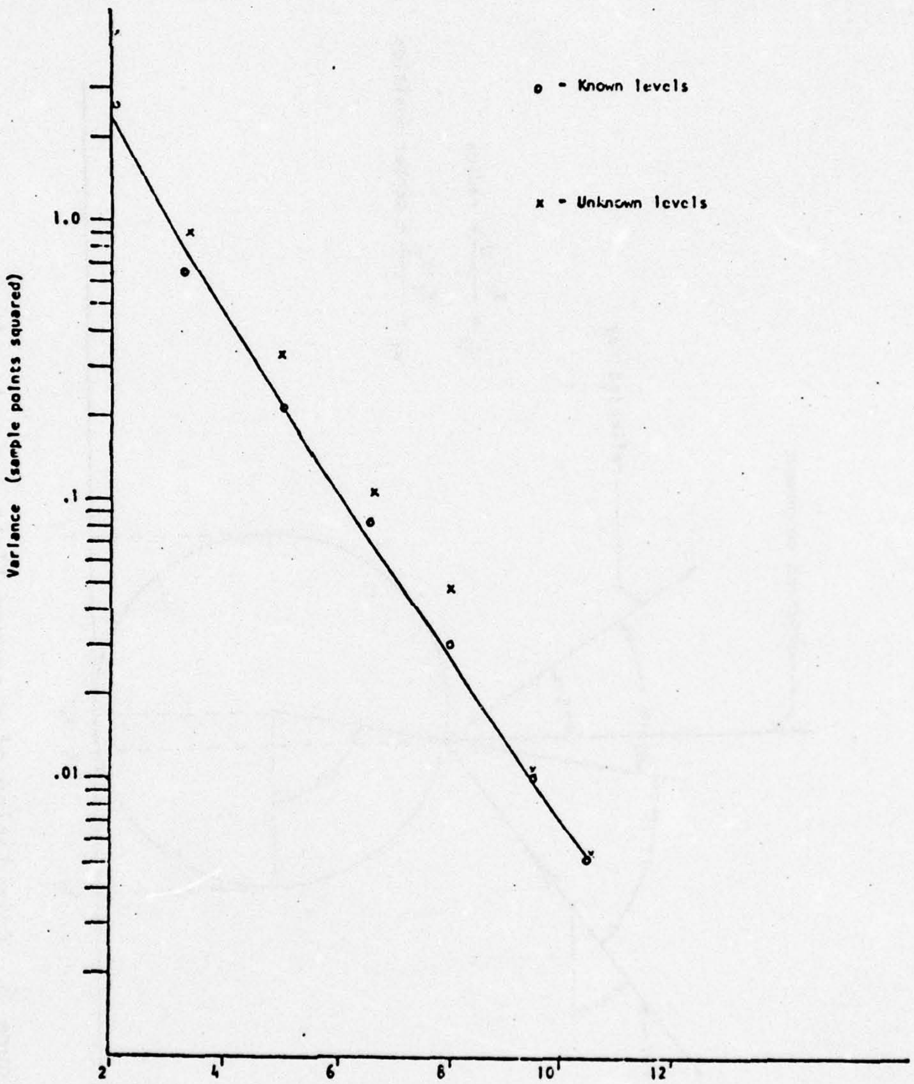


Figure 5 The effect of unknown levels on the variance of the estimate

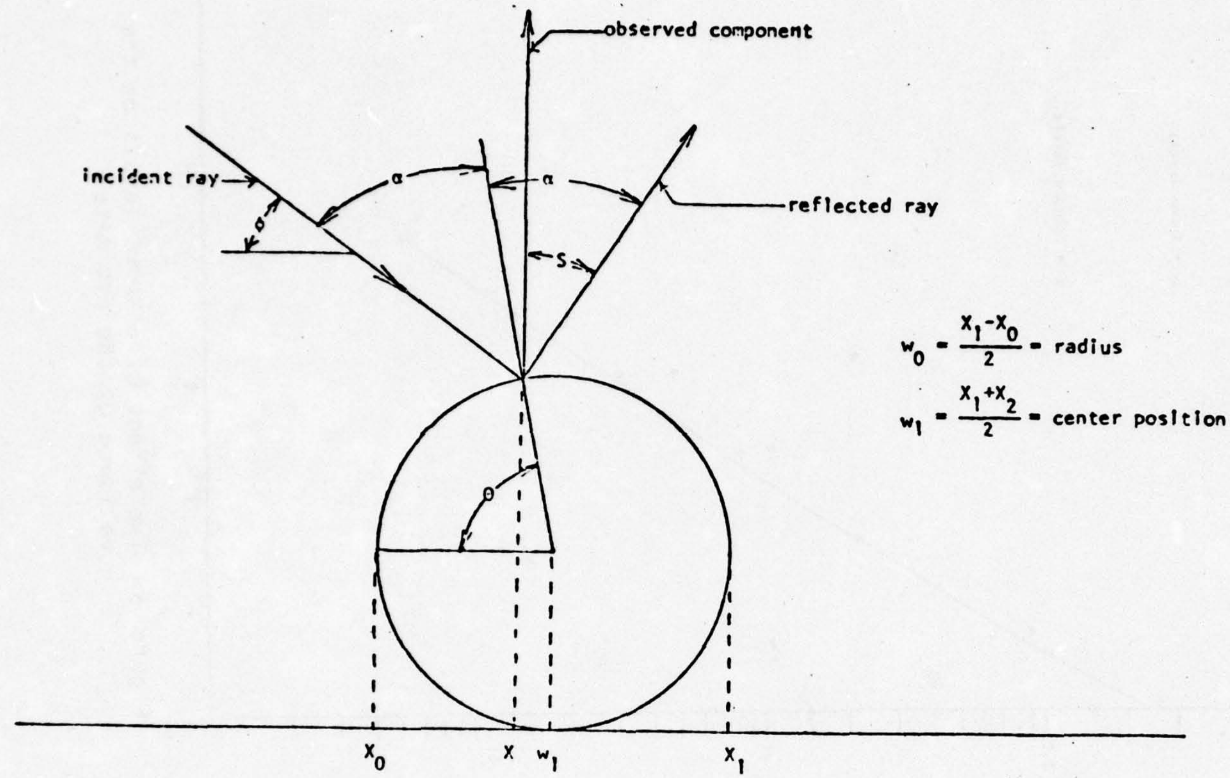


Figure 6 Calculation of observed light intensity from a specularly reflecting cylinder

to the cylinder surface will make an angle θ with the horizontal where (from Figure 6)

$$\cos\theta = \frac{w_1 - X}{w_0}$$

or

$$\theta = \cos^{-1}\left(\frac{w_1 - X}{w_2}\right) \tag{14a}$$

w_1 = center position of the cylinder

w_0 = radius of the cylinder

The angle of incidence α is given by

$$\alpha = \theta - \phi \tag{14b}$$

and

$$S = \theta + \alpha - \frac{\pi}{2} = 2\theta - \phi - \frac{\pi}{2} \tag{14c}$$

$$\text{observed intensity} = CR_S(\alpha)\cos S \tag{14d}$$

A cylinder based on the model of equation (14) was generated and is shown in Fig. 7 A scan line across the cylinder is shown in Fig. 8. The center point w_1 is at position 46, w_0 is 15 sample points and

$$R_S(\alpha) = \begin{cases} .54 - .042\alpha & 0 \leq \alpha \leq 1.3 \\ .4854 + .3276\alpha & 1.3 \leq \alpha \leq \frac{\pi}{2} \end{cases} \tag{15}$$

The cylinder was convolved with a normalized Gaussian line spread function with a standard deviation of two sample points. The blurred intensity samples were converted to density samples with equation (10) and noise with standard deviation given by equation (11) was added. A blurred and noisy scan line is shown in Fig. 9. If the radius was the only unknown quantity a minimum cost

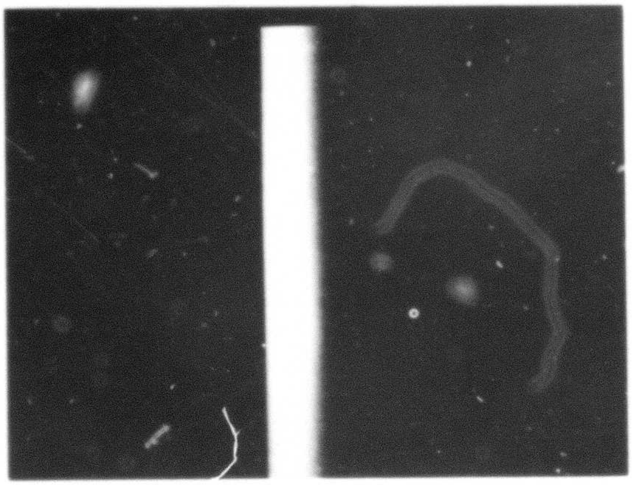


Figure 7 A computer generated cylinder

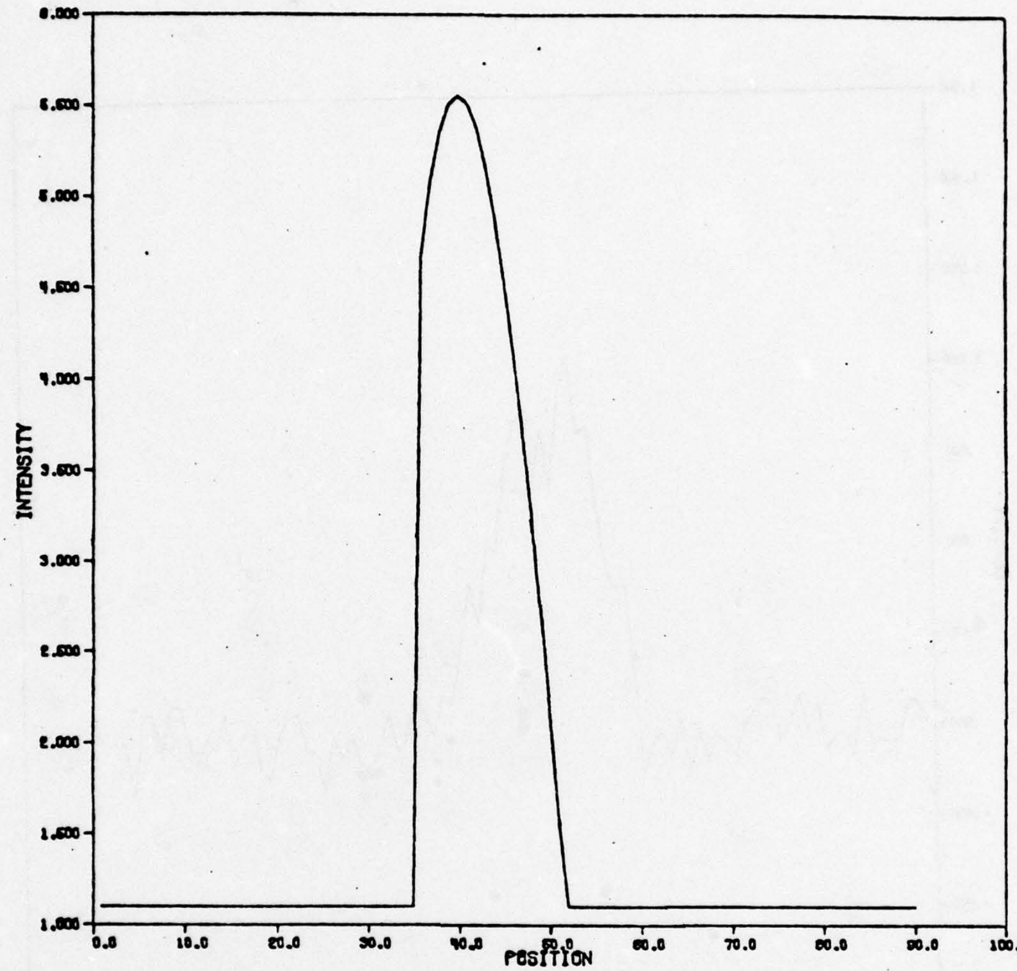


Figure 8 An ideal scan line across a cylinder

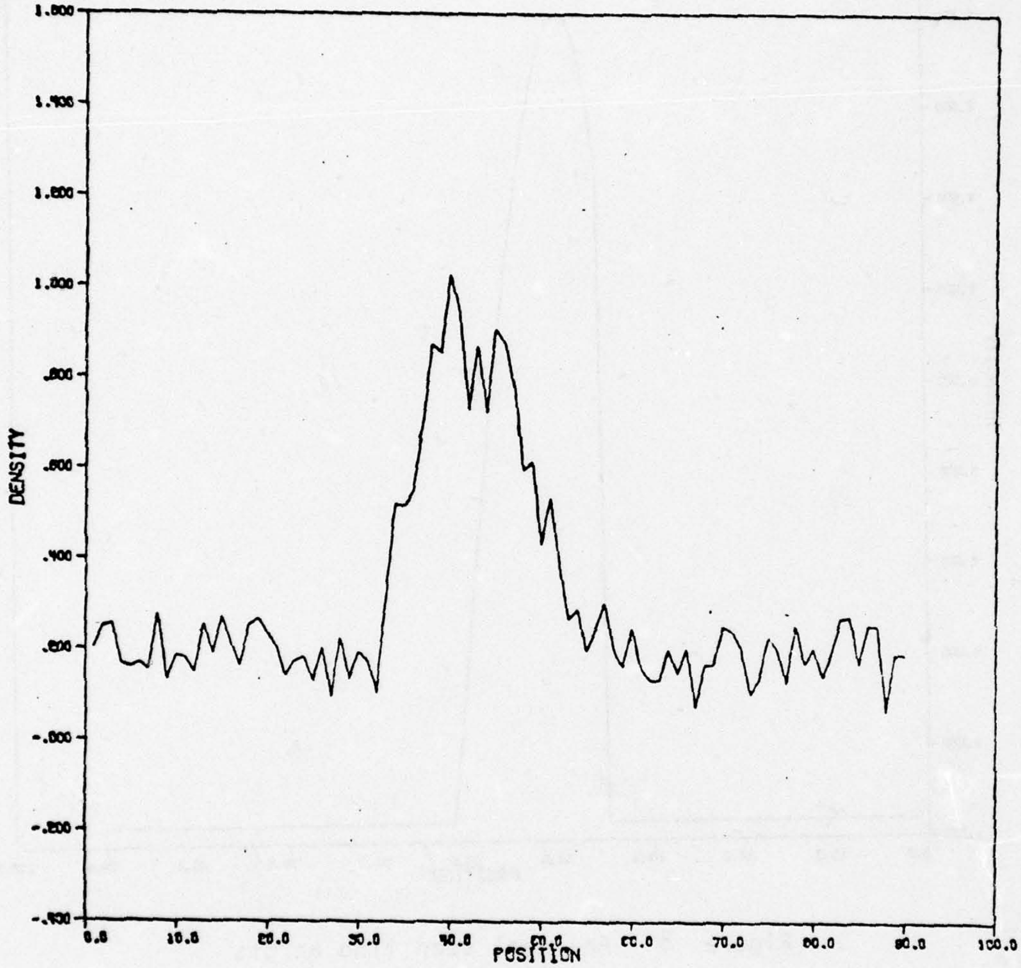


Figure 9 A blurred and noisy scan line
across a cylinder

estimate $\hat{w}_0$ could easily be computed by the VA. However, there is a "nuisance" parameter w_1 which is also assumed unknown.

Assuming the product $CR_S(\alpha)$ is known in equation (14d) an ideal scan line $\underline{l}$ is uniquely determined by the parameters w_0 and w_1. Therefore to maximize

$$p(\underline{Z}|\underline{l})\Big|_{\underline{l}=\hat{\underline{l}}} \tag{16}$$

values of w_0 and w_1 are chosen to maximize

$$p(\underline{Z}|w_0,w_1)\Big|_{\substack{w_0=\hat{w}_0 \\ w_1=\hat{w}_1}} \tag{17}$$

or equivalently to find

$$\min_{w_0,w_1} \sum_{k=1}^{M} \ell n\ p(z_k|w_0,w_1) \tag{18}$$

This minimization can be performed by inspecting the available data to establish ranges in which w_0 and w_1 can be expected to fall, choosing an initial value for w_0 from its possible range, and then finding a value of w_1 (from within its possible range) that minimizes (18). This procedure is repeated with the next possible value for w_0. The costs of the two values for w_0 are compared and the value of w_0 that has the lowest cost is stored as $\hat{w}_0$. Iteration proceeds by finding the value of w_1 that minimizes the cost of the next possible value of w_0. The minimum cost of each possible value for w_0 is compared with the cost of $\hat{w}_0$. If the cost of w_0 is less than the cost of $\hat{w}_0$, w_0 becomes the new $\hat{w}_0$.

Twenty independent measurements of a cylinder radius were made using the technique described above. The results are shown in Table 1. The range of

Table 1
Radius Measurements

True radius	$\hat{w}_0$	Variance of $\hat{w}_0$
15	14.8	.16

possible values for w_0 was taken as ten to twenty sample points and the center position w_1 was assumed to be between sample points forty and fifty. Approximately seventy-five seconds of computer time were required for the twenty measurements.

Measurement of a Road

A 1:5000 scale black and white negative taken with Kodak Plus X Aerographic film was obtained and digitized on a flying spot scanner. The sampling rate was ninety-six samples per millimeter and the data was quantized to 16 bits though only the first 330 levels were occupied. The scene is shown in Figure 10 and shows an intersection of two gravel roads in Warren County, Indiana (the white spot on one of the roads is due to a parity error on a magnetic tape). Figure 11 shows a close-up of one of the roads and Figure 12 shows a scan line across the road of Figure 11. Five hundred training samples from one of the roads showed the average density was .942 with a variance of .00213. One thousand training samples from the field surrounding the road had an average density of .669 with a variance of .00236. The nominal film properties were obtained from Tarkington [4] and Paris [5]. The frequency response of the image blur was assumed to be the product of the film frequency response

$$T_1(f) = \frac{1}{1+(\frac{2\pi f}{250})^2} \tag{19}$$

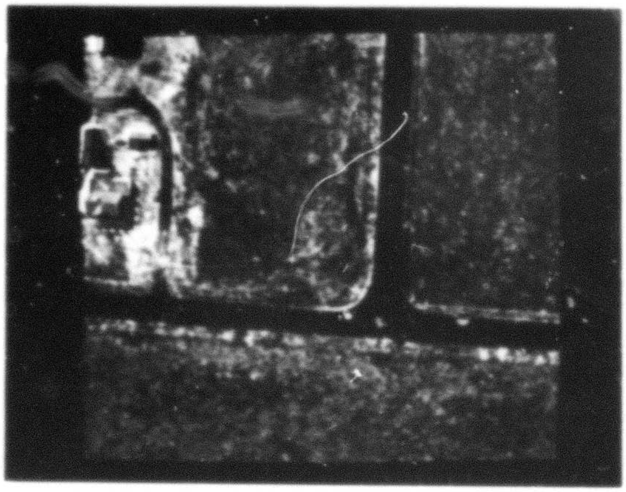


Figure 10 A road intersection in Warren County, Indiana

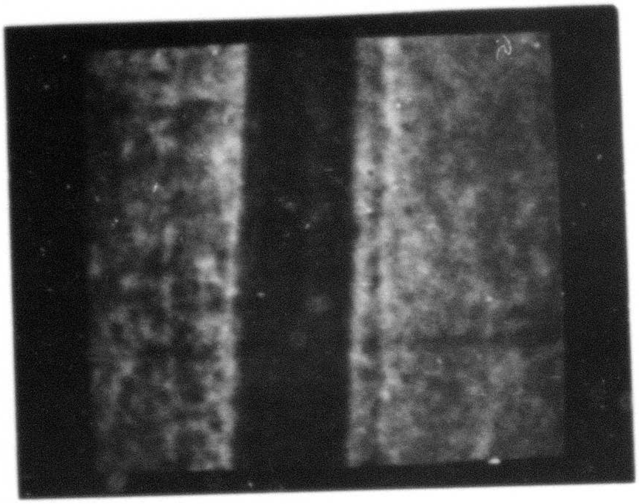


Figure 11 Road Close-up

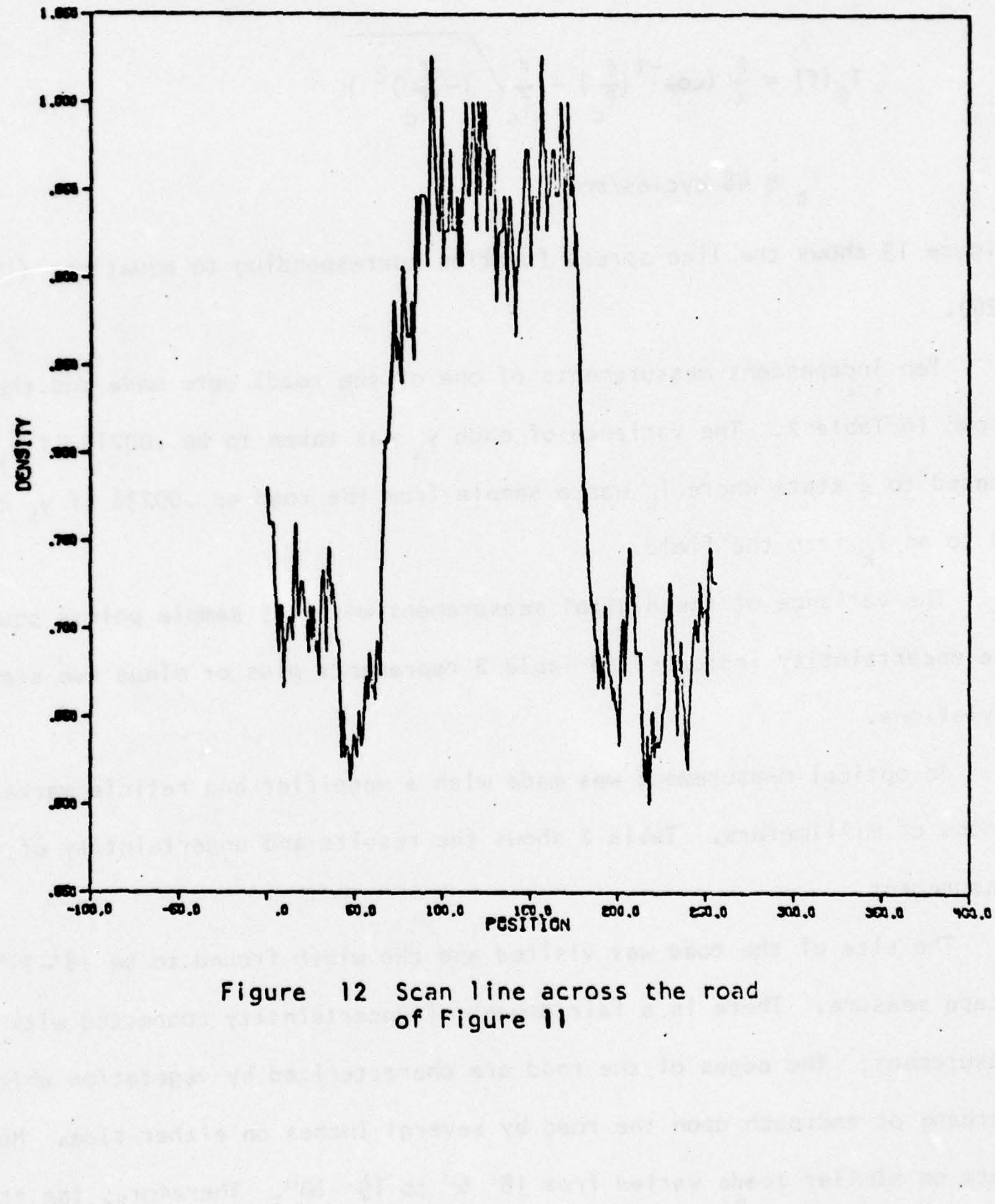


Figure 12 Scan line across the road of Figure 11

and the response of an ideal diffraction limited lens with a cutoff frequency of half the sampling rate:

$$T_2(f) = \frac{2}{\pi} \left(\cos^{-1}\left(\frac{f}{f_c}\right) - \frac{f}{f_c}\sqrt{1-\left(\frac{f}{f_c}\right)^2}\right) \tag{20}$$

$$f_c = 48 \text{ cycles/mm}$$

Figure 13 shows the line spread function corresponding to equations (19) and (20).

Ten independent measurements of one of the roads were made and the results shown in Table 2. The variance of each y_k was taken to be .00213 if y_k corresponded to a state where i_k was a sample from the road or .00236 if y_k corresponded to an i_k from the field.

The variance of the digital measurement was 1.15 sample points squared and the uncertaintity indicated in Table 2 represents plus or minus two standard deviations.

An optical measurement was made with a magnifier and reticle marked in tenths of millimeters. Table 2 shows the results and uncertaintity of this measurement.

The site of the road was visited and the width fround to be 18'-11" with a tape measure. There is a fair amount of uncertaintity connected with this measurement. The edges of the road are characterized by vegetation which can overhang or encroach upon the road by several inches on either side. Measurements on similar roads varied from 18' 6" to 19' 10". Therefore, the true width of the road the day the photograph was taken is not known exactly.

The Cramer-Rao bound σ^2_{CR} was calculated assuming the density levels, variances and line spread function used by the VA were correct. This variance was 1.23 sample points squared which is reasonably consistent with σ^2_{VA}.

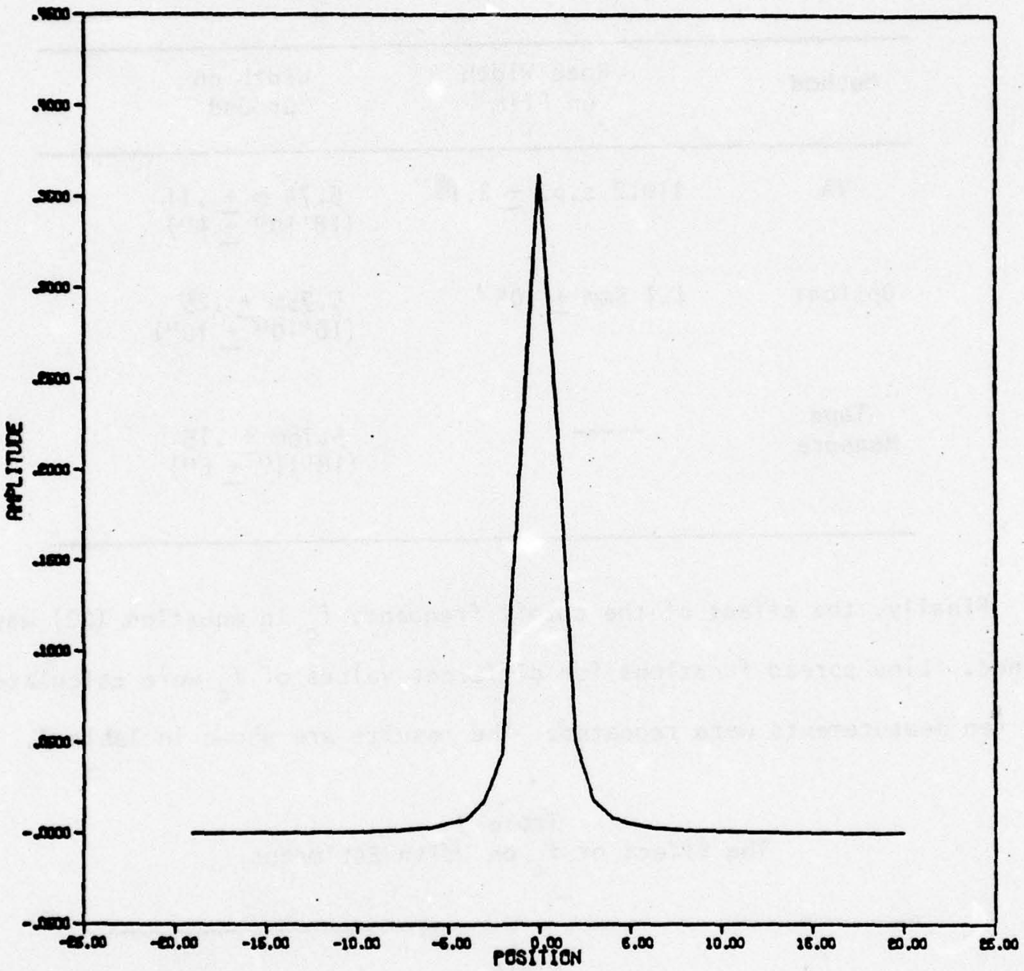


Figure 13 The line spread function of the film-lens combination

Table 2
Road Width Measurement Results

Method	Road Width on Film	Width on Ground
VA	110.2 s.p. ± 2.1	5.74 m ± .11 (18'10" ± 4")
Optical	1.1 5mm ± .05	5.75m ± .25 (18'10" ± 10")
Tape Measure	-----	5.76m ± .15 (18'11" ± 6")

Finally, the effect of the cutoff frequency f_c in equation (20) was examined. Line spread functions for different values of f_c were calculated and the ten measurements were repeated. The results are shown in Table 3.

Table 3
The Effect of f_c on Wdith Estimates

f_c (cycles/mm)	Width (sample points)	Variance
56	110.3	1.20
48	110.2	1.15
40	109.8	1.1
32	109.1	.96

Table 3 indicates the width estimates produced by the VA are not overly sensitive to imperfect knowledge of the degrading system.

REFERENCES

[1] J. Burnett and T. S. Huang, "Application of Finite State Markov Process Model to Image Measurement", in Image Understanding and Information Extraction QPR February-April 1976.

[2] H. Van Trees, Detection, Estimation and Modulation Theory, Part I, Wiley, N.Y, 1968.

[3] R. Gonsalves, "Cramer-Rao Bounds on Mensuration Errors," Applied Optics, Vol. 15, May 1976, pp. 1270-1275.

[4] R. Tarkington, "Kodak Panchromatic Negative Films for Aerial Photography," Photogrammetric Engr., December 1959, pp. 695-699.

[5] D. Paris, "Approximation of the Sine Wave Response of Photographic Emulsions," JOSA, Vol. 51, No. 9, Sept. 1961, pp. 988-991.

[6] Tuong-Phong, "Illumination for Computer Generated Images," Univ. of Utah Technical Report CSC-73-129, July 1973.

IMAGE SEGMENTATION BY CLUSTERING

M. Y. Yoo and T. S. Huang

I. Introduction

Two-dimensional photographic images consist of several fundamental pictorial components. Each component has a more-or-less different property or characteristic to human visual perception. We may roughly call these components textural components or simply textures. It is almost impossible to describe the textures precisely because of the abundant variety of them in the real world, whereas it is highly desirable to have even a rough measure to distinguish these textural components.

Zucker [1] tried to model the textures based on the concept of "primitive" and regular or quasi-regular patterns, but his approach is far from being practical. A more practical approach to texture is Haralick's [2] application of textural features for image classification based on the spatial dependence matrix.

The set of possible descriptions of a picture is often so large that it is impractical to describe the picture by assigning it to an element of this set. Instead, a more practical description may often be given by partitioning the picture into objects and assigning each of these objects to one element of a set of possible descriptions of objects.

Segmentation is the partitioning of images into several basic textural components, each of which has significantly different properties, (statistical, topological). There have been several approaches in this direction. Fisher [3] tried to partition the picture function into a "unimodal subset" which means conceptually a subset having only one "hill" in the intensity values of the points in the subset and Gupta [4] and Kettig [5] adopted the statistical hypothesis-testing of local mean and variance to detect the boundaries in closed forms and applied this approach to data compression and classification.

The human visual system is an excellent textural discrimination and Julesz's [6] experiments show that not only the statistical but also the topological properties of images are important factors to textural discrimination. So the best textural discriminator is the combination of statistical measures and topological properties of images. Topological properties usually are described by either sytactic methods or some algebraic measures.

We are not at this point in a position to combine the statistical description with the syntactic description in an appropriate way; for the present we are mainly concerned with the pure statistical or the pure algebraic approach. The approach which we propose consists of extracting pair features using a 3×3 moving window, "eyeball clustering" of features and back-transformation of the feature plane onto the original picture domain. This approach is motivated by the different types of pair feature distributions for 16 different textures shown in Brodatz [7]. (See Fig. 1 for four examples; the horizontal axis is sample standard derivation and the vertical axis denotes sample mean. The size of the feature plane is 64×64.)

II. The Image Segmentation Algorithm

The image segmentation algorithm which we propose consists of three major steps: (1) feature pair extraction, (2) clustering of features, and (3) segmentation. The feature extraction is the most important step and is the extraction of a certain measure which represents the local characteristics of the image is a reasonably simple form.

The clustering of features is quite dependent upon the features chosen in the first step. If ideal features were chosen, the features are well clustered in the feature plane and clustering is trivial, otherwise some heuristic clustering algorithm or "eyeball clustering" should be adopted. The segmentation is the back-transformation of the clustered features onto the original picture domain in the segmented image form. The way of presentation of segmented pictures may be either displaying the boundaries between different textural components of the image or display of different major textures in separate picture domain.

In many applications (data compression, shape description, classification, etc.) the location of boundaries between different textures is important information so we have chosen the display of boundaries.

The detailed description of the three steps will follow.

A. The Feature Extraction

The extraction of features is a very important part in image segmentation. The best features may be the detailed description of the structural relationship between the selected picture array and its surroundings, but this is very complex to be implemented for computer processing. A reasonable measure which is significantly simplified but still contains major information about the selected picture array is the statistical or the structural characteristics within a window of appropriate size centered within the array. We may lose some information by this simplification but in some sense this approach is more reasonable than detailed information for the textural discrimination. Natural images usually consist of many or several textures which are smoothly varying in shape within the same pattern rather than strictly webbed and every picture array may be identified as a different texture by the detailed description.

Multidimensional features require significantly increased memory size in the computation and it is also impossible to see the clustering in the hyper-feature space so we have worked with feature pairs. The three feature pairs which we have tried are the following:

1. The local sample mean and the local sample standard deviation.

```
(x-m,y-n)                    (x-m,y+n)
 - - - - - - - -   - - - - - - - - -
                  n
 - - - - - - - -   - - - - - - - - -
         m      (x,y)

 - - - - - - - -   - - - - - - - - -
(x+m,y-n)                    (x+m,y+n)
```

Figure 2 Local window used for local operations

Let P(x,y) be the picture array. Then the local sample mean M(x,y) and the local sample standard deviation S(x,y) at the array point (x,y) based on a local window size m x n are:

$$M(x,y) = \frac{1}{(2m+1)(2n+1)} \sum_{\mu=x-m}^{x+m} \sum_{\sigma=y-n}^{y+n} P(\mu,\sigma)$$

$$S(x,y) = \frac{1}{(2m+1)(2n+1)} \sum_{\mu=x-m}^{x+m} \sum_{\sigma=y-n}^{y+n} \{P(\mu,\sigma) - M(x,y)\}^2 \quad 1/2$$

2. Local minimum and local maximum:

Let $D = \{(\alpha,\beta): \alpha = x-m, \ldots, x+m, \beta = y-n, \ldots, y+n\}$ then the local minimum and the local maximum are

$$MIN\ (x,y) = \underset{(\alpha,\beta)\,\varepsilon\, D}{\text{minimum}}\ P(\alpha,\beta)$$

$$MAX\ (x,y) = \underset{(\alpha,\beta)\,\varepsilon\, D}{\text{maximum}}\ P(\alpha,\beta)$$

3. The number of the local "jumps" and the average magnitude of the jump:

We compare two adjacent points (in all directions) in the local window and if the grey level change is greater than a preassigned threshold value we assume there is a jump. When we count the number of "jumps" we consider the "jump ups" and "jump downs" and two successive "jump ups" or "jump downs" are counted as one jump with a large amplitude. We take the total number of jumps in the local area as feature 1 and the average magnitude of a jump as feature 2.

feature 1 = total number of jumps in the local image of size $m \times n$

feature 2 = the average magnitude of a jump in the area

B. The Clustering of Features

There are many different approaches [8-12] reported for clustering data. A simple and practical approach is ISODATA (Iterative Self-Organizing Data Analysis Technique [9]). In this algorithm several important initial clustering centers are picked up and assigned levels. Each sample point is merged into the nearest center based on the Euclidean distance. Based on the initial grouping, new clustering centers are calculated and if the new clustering centers are the same as the old ones the clustering process is terminated; otherwise the same kind of merging process is repeated.

The Euclidean distance may be modified if the extracted features have significantly different magnitudes to avoid the "masking effect" of the dominant feature, as follows:

$$d = \sqrt{W_1 (M_2 - M_1)^2 + W_2 (\sigma_2 - \sigma_1)^2}$$

Where M_i, σ_i are the mean and standard deviation at each sample point. The W_i's are weights. ISODATA is easy to implement but does not give smooth boundaries in the feature plane.

The graph theoretic approach is a more elaborate clustering algorithm [12]. For large sample size this algorithm becomes basically the same as the valley seeking approach. The algorithm gives smooth boundaries but requires an extremely large memory size for two dimensional data sets and is not practical for large samples (64×64 is the largest data size for practical application but further reduction of data is necessary for efficient use). "Eyeball clustering" is the most accurate and is very flexible because we have complete control of the data. We look at the data and cluster them in arbitrary groups based on our previous experience. "Eyeball clustering" is used for this experiment because automatic clustering is practically impossible by the ISODATA or graph theoretic clustering technique for very large images.

C. The Segmentation

This is the back-transformation of clusters in the feature plane onto the textural components in the original picture domain. Different clusters in the feature plane correspond to different textures in the original image domain and the number of textural components depends upon how many clusters we allow in the feature plane. An alternate way of segmenting images is locating boundaries between different textures.

Resulting boundaries form closed contours except minor isolated or clustered noisy points when we set boundaries between different textural components. Once the clustering of features is complete, the textural discrimination in the picture domain is determined and how we transform the clusters back onto the picture plane does not affect the locations of the textural components. Therefore, there is no preferred direction in texture discrimination processing (compare with the BLOBS [13]). More clusters in the feature plane give finer boundaries in the picture plane and this algorithm is a type of parallel top-to-bottom approach.

III. Experimental Results

We calculate feature pairs at each picture array point using a 3 x 3 moving window centered at the array and normalize and quantize features to appropriate levels. The number of the local jumps is a fairly small integer and does not have to be quantized, but other features are quantized to 128 levels such that the maximum levels of a feature is 128.

Computer plots of the feature planes are shown in Fig. 3 to Fig. 10. Only major clusters show up in the computer line print outs and our "eyeball clustering" is based on those of the line print outs, and these gould electrostatic prints are used in conjunction with line printer outputs of the same information to locate the initial cluster. Typical line printer output corresponding to Figs. 3 and 4 are shown in Figs. 11 and 12.

The density in the feature plane denotes the total number of picture points in the original image plane which have certain feature pairs corresponding to the coordinates of the feature plane. If there is a dense cluster (Fig. 3) in the feature plane which may correspond to a large textural component of the original picture, some other small clusters corresponding to small textural components do not show up in the feature plane and renormalization of data excluding the data contributing to the dense cluster is necessary to see minor clusters (Fig. 4). Actually in many cases, the feature plane has several clusters which give major boundaries in the pictorial plane.

The segmentation based on the local jump is not quite adequate for the pictures "Girl" and "Professor" and we did not include them in this report. Sample clustering regions are indicated in Figs. 13 and 14. The regions shown in Fig. 13 are based on the clusters in Fig. 11. The regions in Fig. 14 are based on the combination of the original dense cluster (Fig. 11) and the sub-clusters (Fig. 12). Each clustering level in each feature plane was transformed back into the original picture domain and the corresponding picture array is given the same level. So different textures in the picture domain have different levels corresponding to the levels of the feature plane. We arbitrarily set a boundary on the picture point with the higher level when two different levels are adjacent. Boundaries of three different pictures (missile, girl, professor) based on three different feature pairs are shown in part (A) of Figs. 16 to 19 and Figs. 21 and 22, and Figs. 24 and 25. Part (B) of Figs. 16 to 19 show the over display of the boundaries on the original picture and gives some idea about the accuracy of the algorithm.

The boundaries shown in Fig. 17(A) are the combination of the major boundaries (see Fig. 16(A)) and the boundaries detected based on renormalized subfeatures (see Fig. 4 or 12). The single isolated noisy points are dropped in all cases except in Figs. 16, 18, and 19. For Fig. 19 we used a threshold value of 30 when we calculated the feature pairs.

References

1. S. W. Zucker, "Toward a Model of Texture," Computer Graphics and Image Processing, No. 5, pp. 190-202, June 1976.

2. R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural Features for Image Classification," IEEE Trans. on Systems, Man, and Cybernetics, Vol. SMC-3, No. 6, pp. 610-621, November 1973.

3. R. L. Walton and P. D. Fisher, "A Picture Segmentation Algorithm and Its Application to Clustering," Milwaukee Symposium on Automatic Control, pp. 40-47, 1974.

4. J. N. Gupta and P. A. Wintz, "A Boundary Finding Algorithm and Its Applications," IEEE Trans. on Circuits and Systems, Vol. CAS-22, No. 4, April 1975.

5. R. L. Kettig and D. A. Landgrebe, "Classification of Multispectral Image Data by Extraction and Classification of Homogeneous Objects," Proc. Purdue Symposium on Machine Processing of Remotely Sensed Data, pp. 2A-1-2A-11. 1975.

6. B. Julesz, "Visual Pattern Discrimination," IRE Trans. on Information Theory, pp. 84-92, Feb. 1962.

7. P. Brodatz, Texture, Dover Publications: New York, 1966.

8. H. P. Friedman and J. Rubin, "On Some Invariant Criteria for Grouping Data," JASA, pp. 1159-1178, December 1967.

9. G. H. Ball and D. J. Hall, "A Clustering Technique for Summarizing Multivariate Data," Behavioral Science, Vol. 12, pp. 153-155, March 1967.

10. K. Fukunaga and W. Koontz, "A Criterion and an Algorithm for Grouping Data," IEEE Trans. on Computers, Vol. C-19, No. 10, pp. 917-923, Oct. 1970.

11. R. A. Jarvis and E. A. Patrick, "Clustering Using a Similarity Measure Based on Shared Near Neighbors," IEEE Trans. on Computers, Vol. C-22, No. 11, pp. 1025-1034, November 1973.

12. W. Koontz, P. M. Narendra, and K. Fukunaga, "A Graph Theoretic Approach to Non-parametric Cluster Analysis," IEEE Trans. on Computers, Vol. C-25, No. 9, pp. 936-944, September 1976.

13. J. N. Gupta and P. A. Wintz, "Multi-Image Modeling," School of Electrical Engineering, Purdue University, West Lafayette, IN, Technical Report, TR-EE 74-24, 1974.

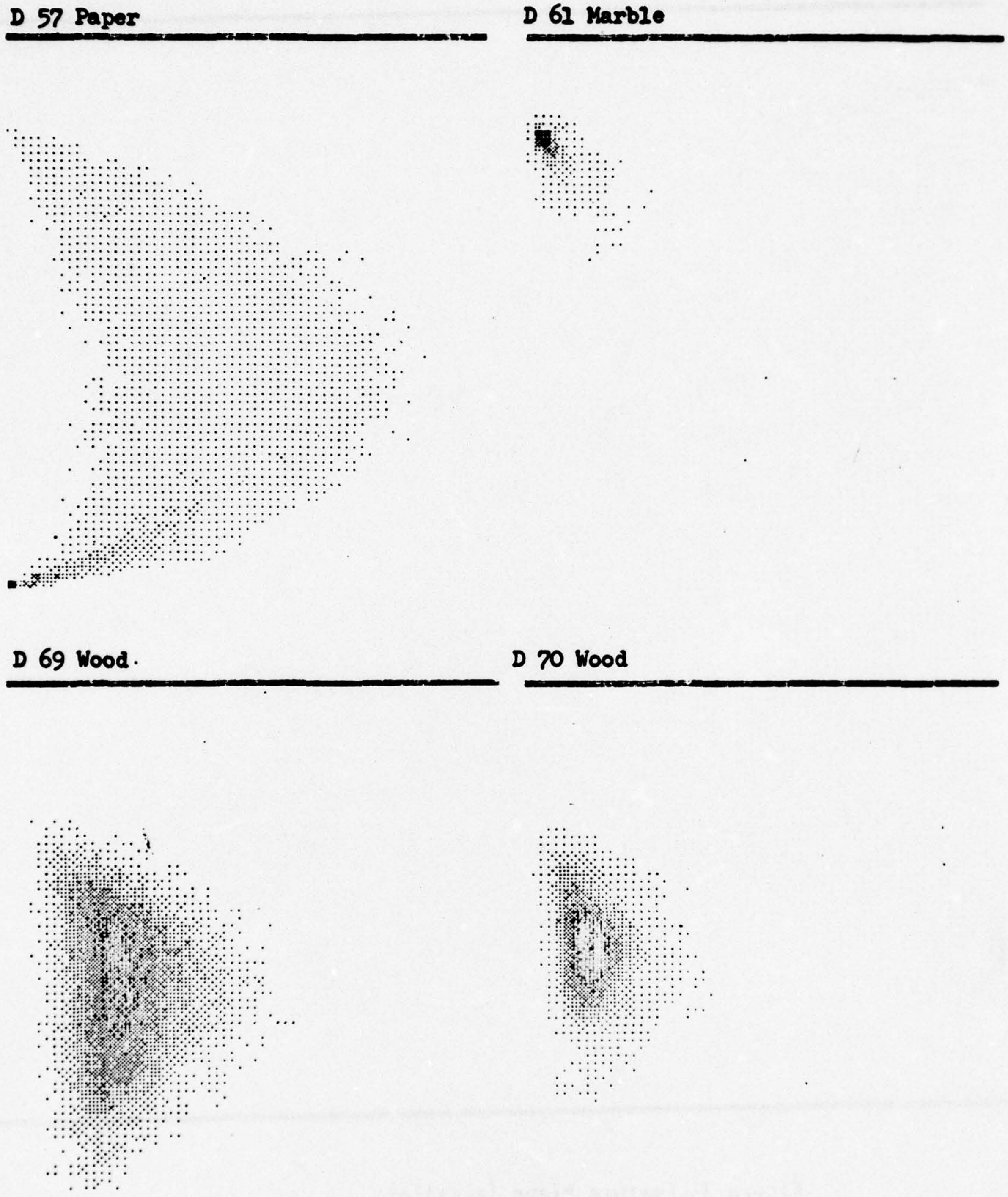


Figure 1 Feature plane for sample textures.
Abcissa is standard deviation.
Ordinate is mean.

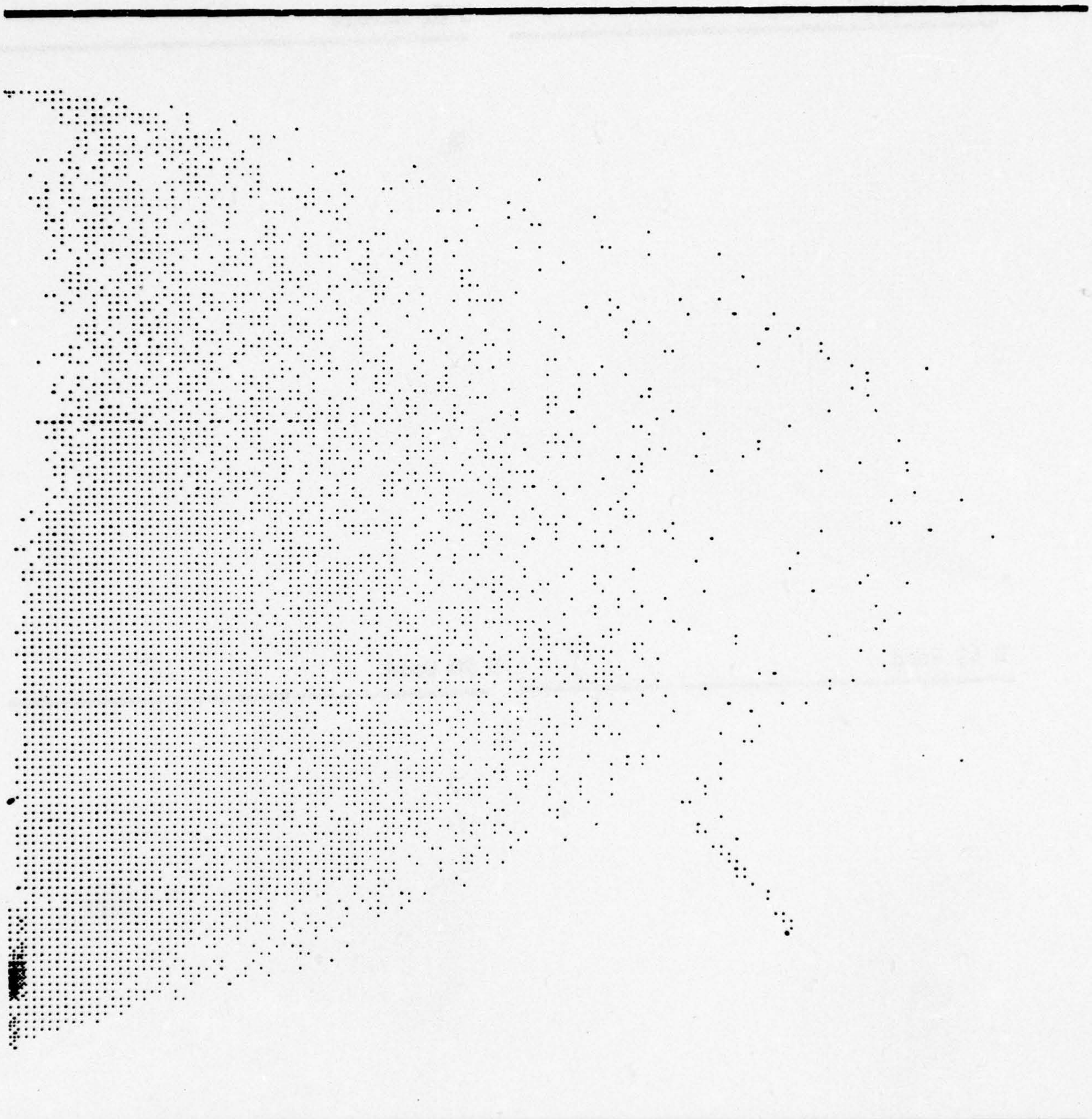


Figure 3 Feature Plane (missile)
Horizontal Axis: Standard Deviation
Vertical Axis: Mean
Size: 128 x 128

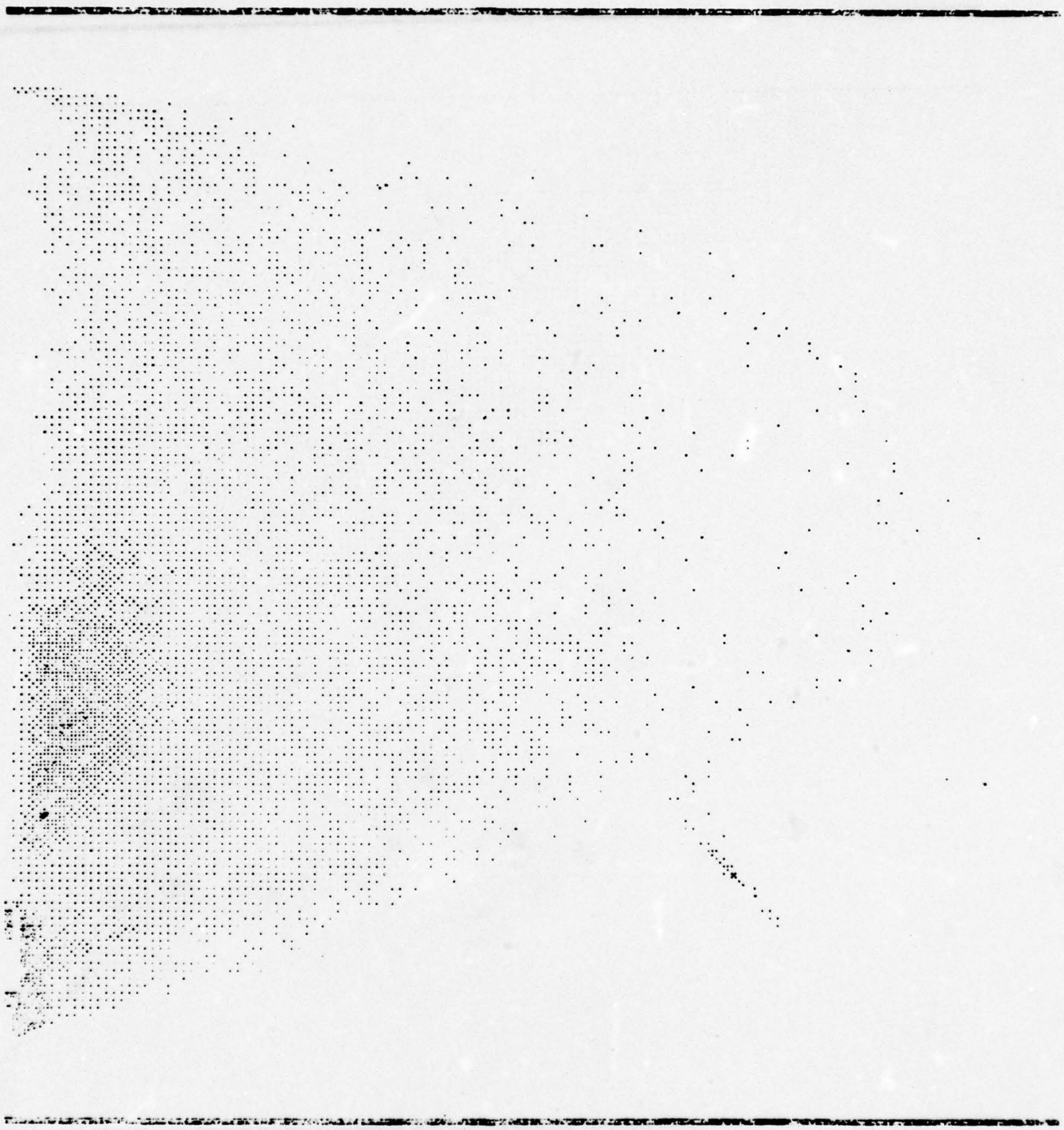


Figure 4 Sub Feature Plane (missile)
Horizontal Axis: Standard Deviation
Vertical Axis: Mean
Size: 128 x 128

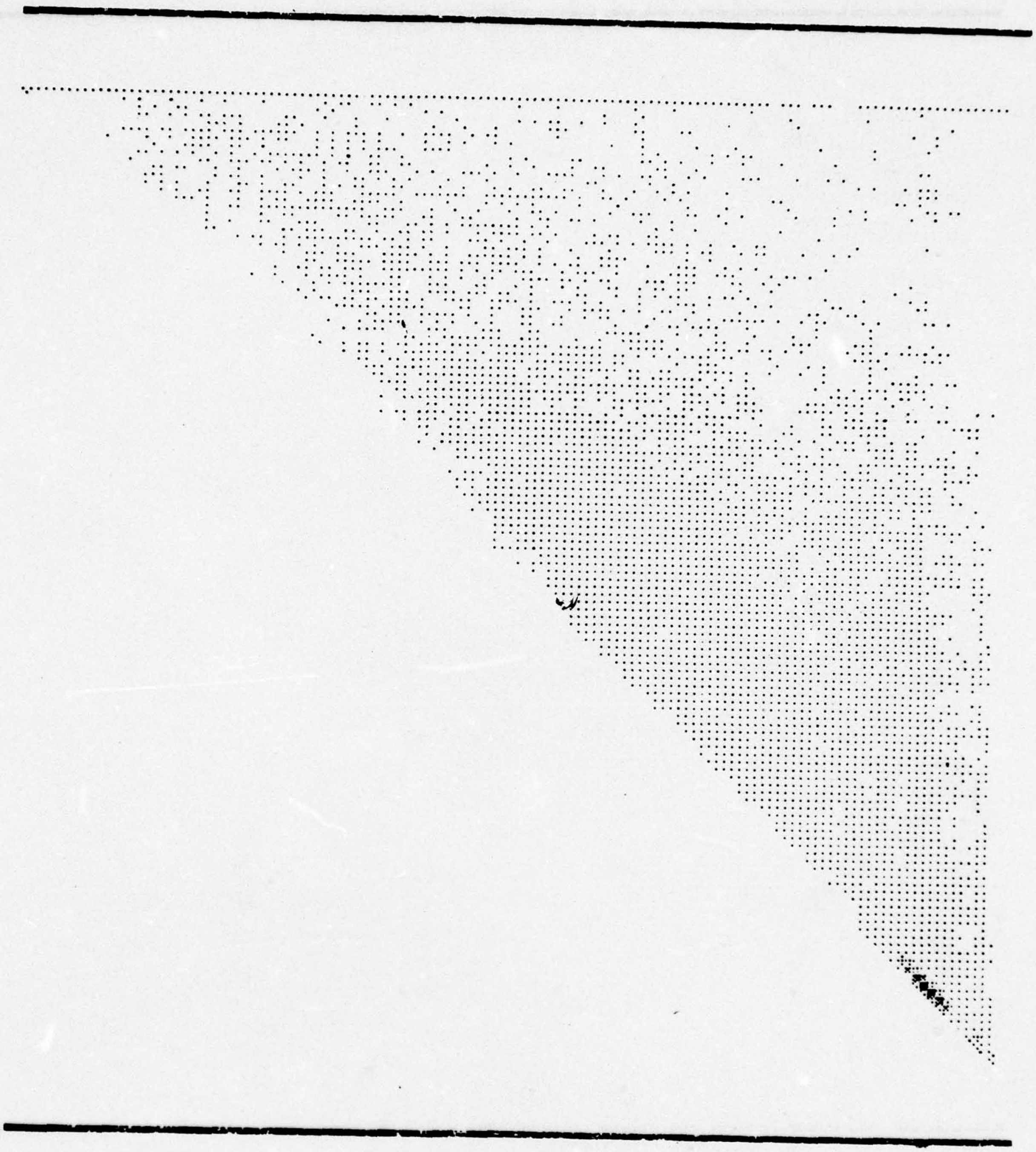


Figure 5 Feature Plane (missile)
Horizontal Axis: Local Maximum
Vertical Axis: Local Minimum
Size: 128 x 128

Figure 6 Feature Plane (missile)
Horizontal Axis: Number of local jump
Vertical Axis: Average amp of the jump
Size: 20 x 128

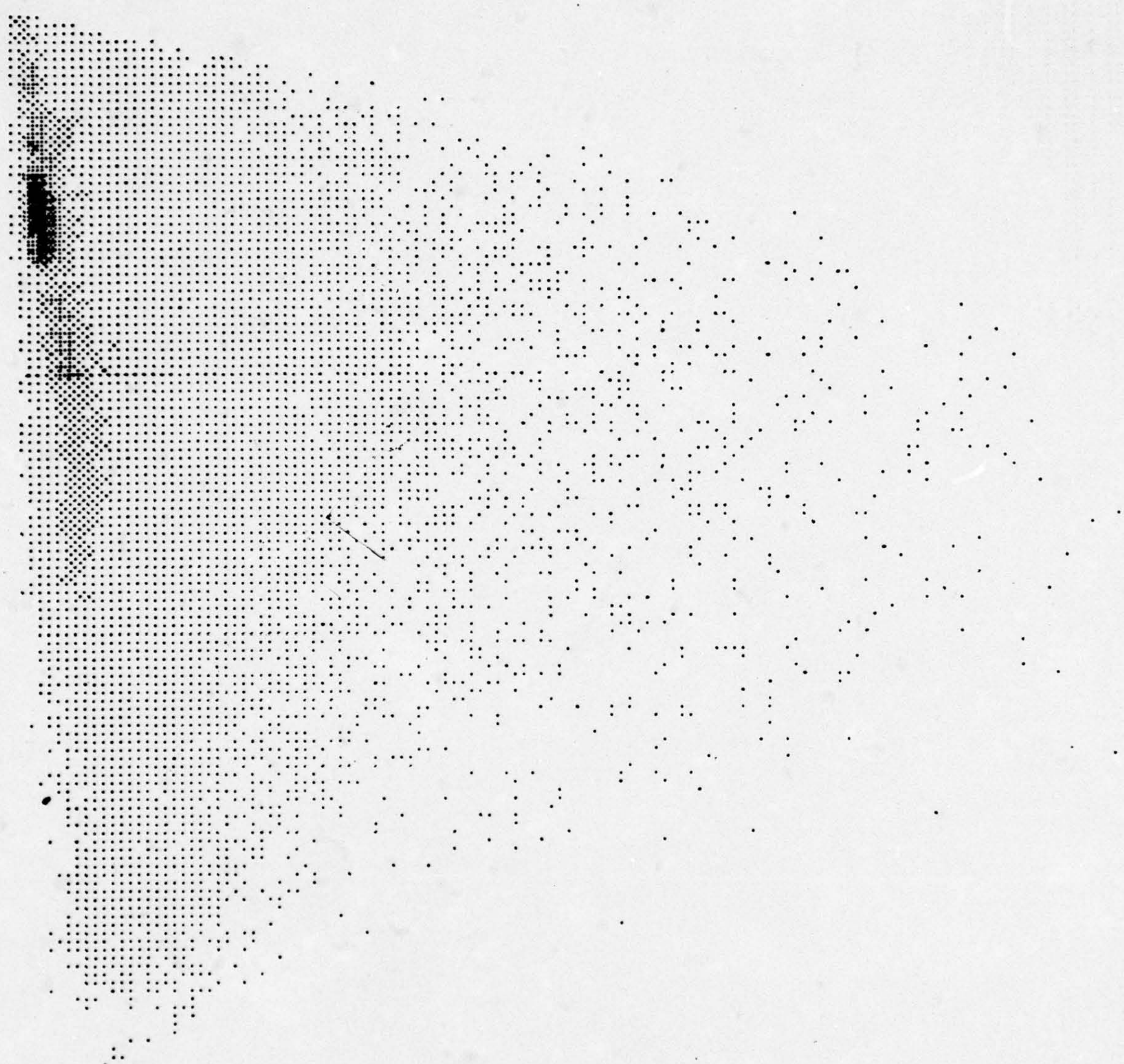


Figure 7 Feature Plane (girl)
Horizontal Axis: Standard Deviation
Vertical Axis: Mean
Size: 128 x 128

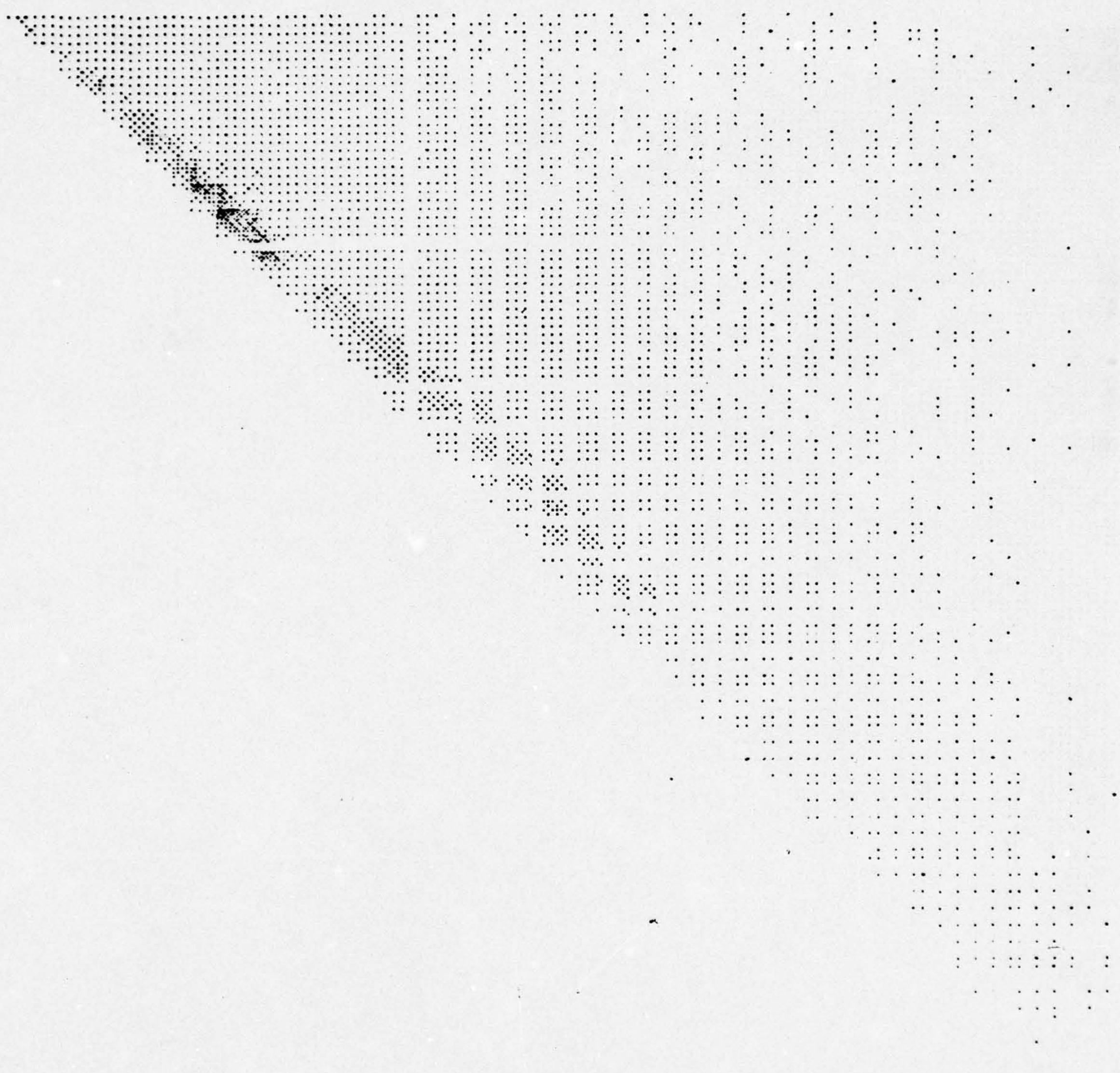


Figure 8 Feature Plane (girl)
Horizontal Axis: Local Maximum
Vertical Axis: Local Minimum
Size: 128 x 128

Figure 9 Feature Plane (professor)
Horizontal Axis: Standard Deviation
Vertical Axis: Mean
Size: 128 x 128

Figure 10 Feature Plane (professor)
Horizontal Axis: Local Minimum
Vertical Axis: Local Maximum
Size: 128 x 128

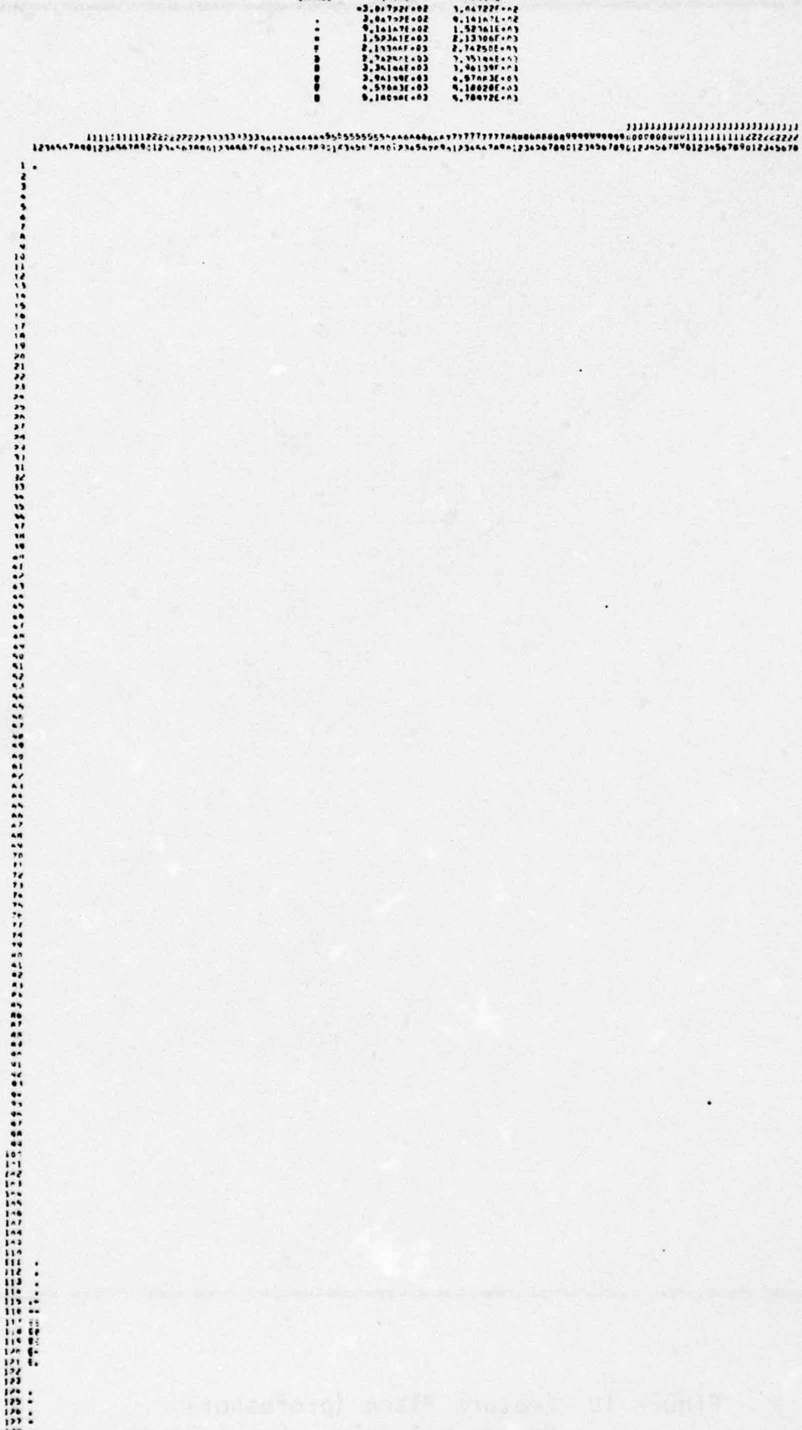


Figure 11 Feature Plane (missile)
Horizontal Axis: Standard Deviation
Vertical Axis: Mean

Figure 12 Sub Feature Plane (missile)
Horizontal Axis: Standard Deviation
Vertical Axis: Mean

MISSILE

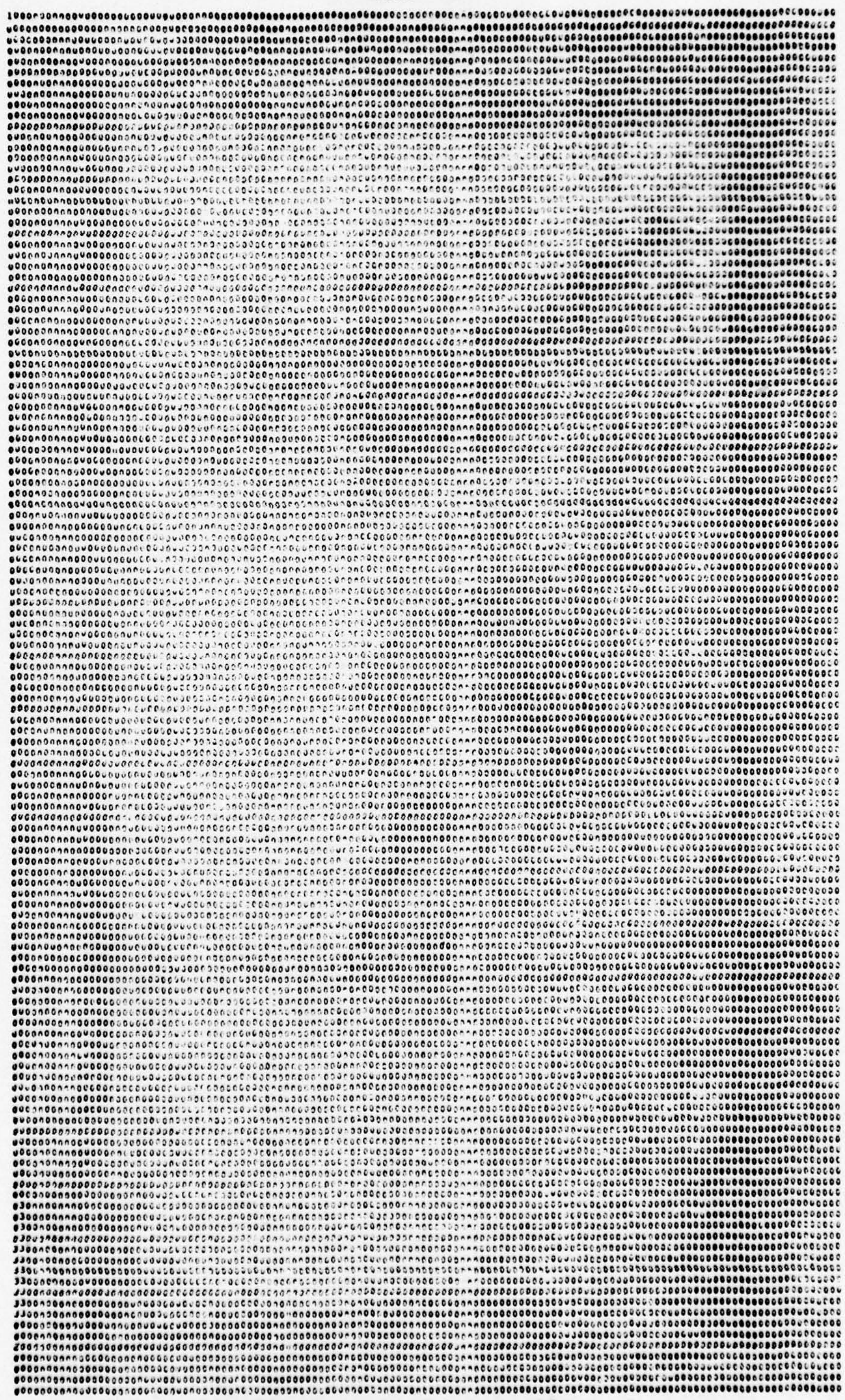


Figure 13 Clusters in Feature Plane
Horizontal Axis: Standard Deviation
Vertical Axis: Mean
Size: 128 x 128

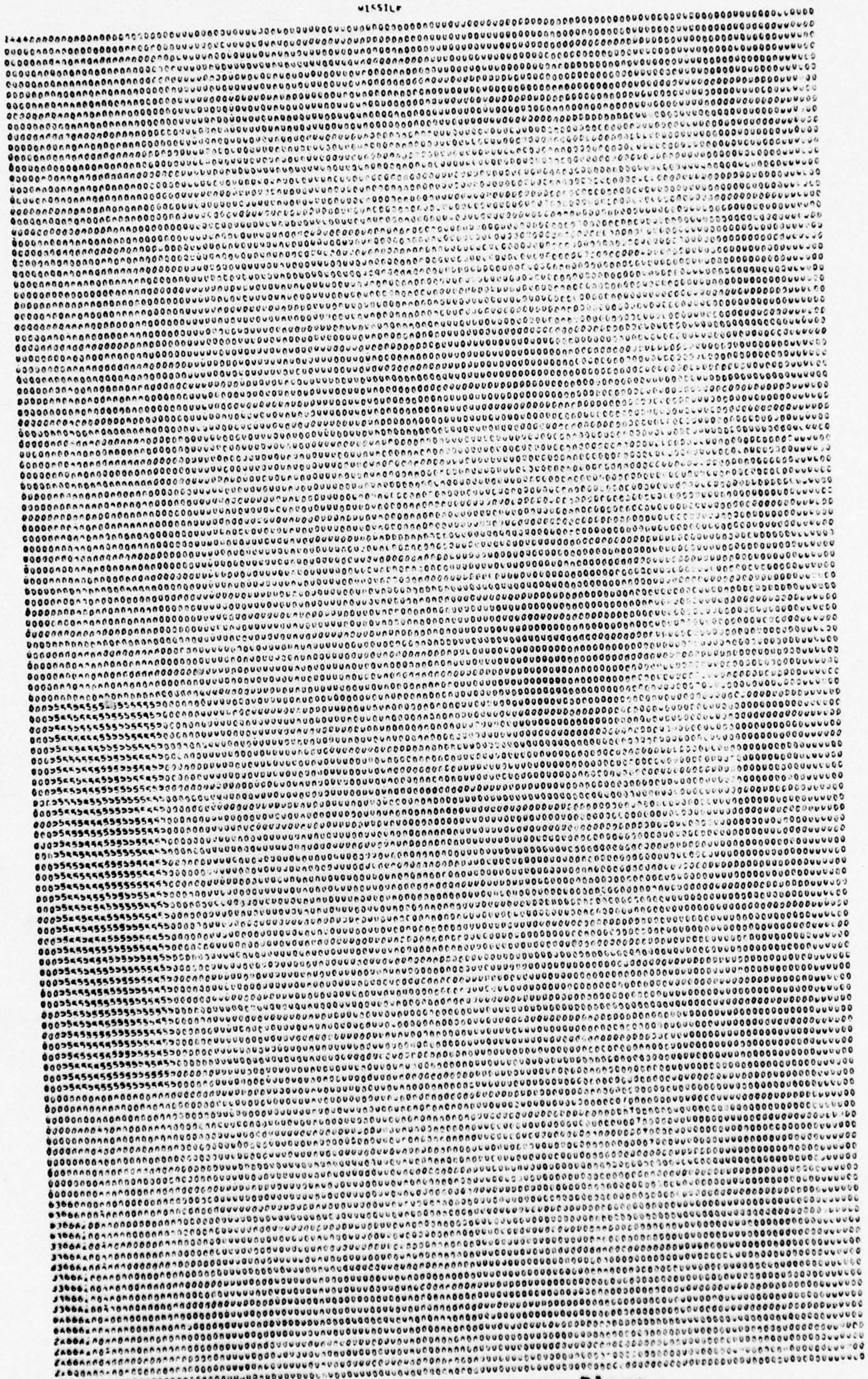


Figure 14 Clusters in Feature Plane
Horizontal Axis: Standard Deviation
Vertical Axis: Mean
Size: 128 x 128

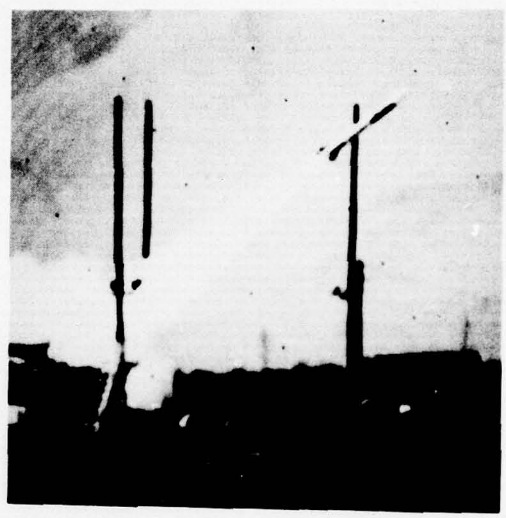
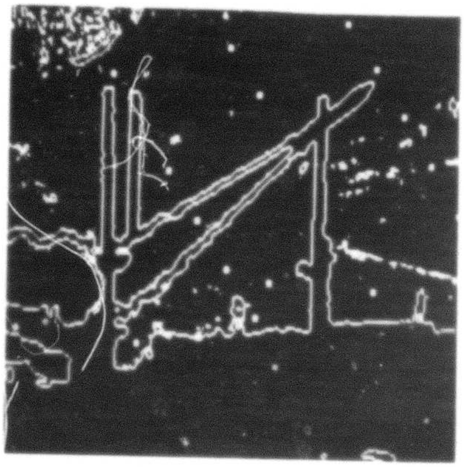
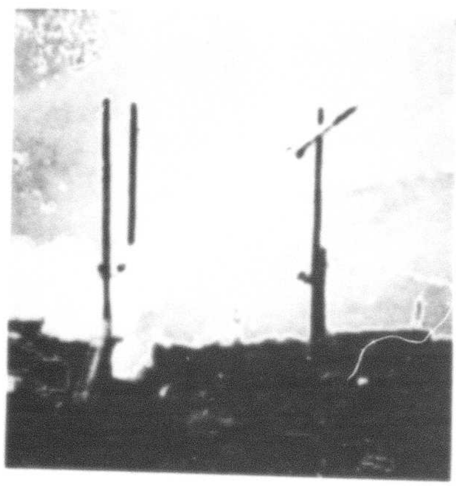


Figure 15 The original picture.
Size: 256 x 256

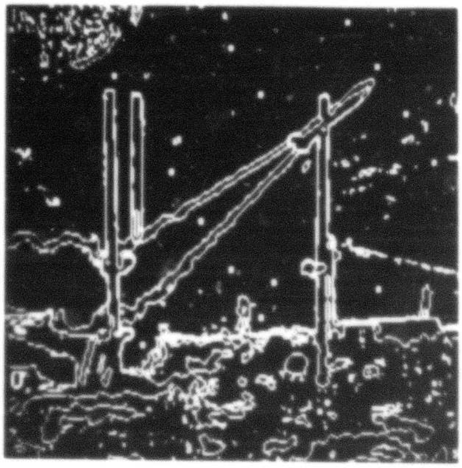


(A) Boundary

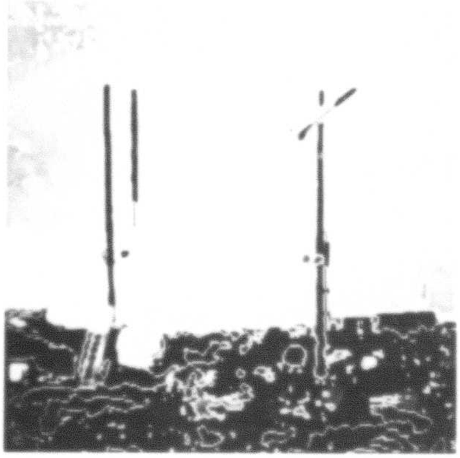


(B) Segmentation

Figure 16 The segmented picture. Mean and standard deviation used.

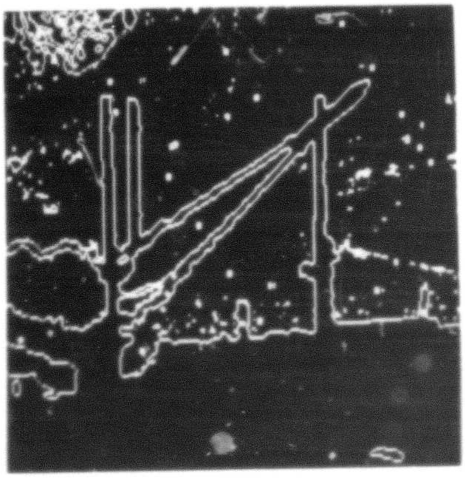


(A) Boundary

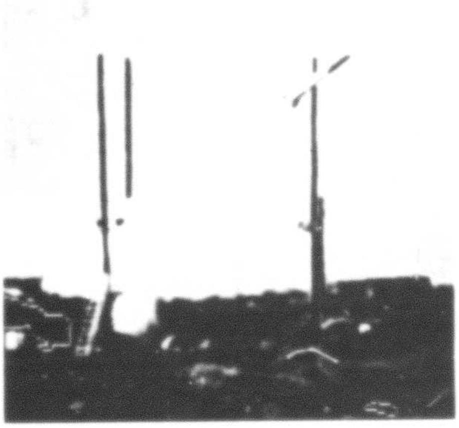


(B) Segmentation

Figure 17 The segmented picture.
Mean and standard
deviation used.
Subfeature used.

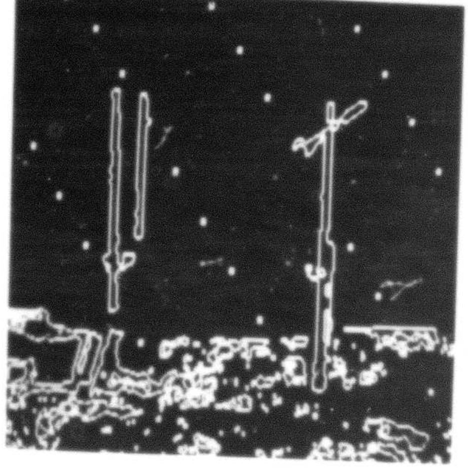


(A) Boundary

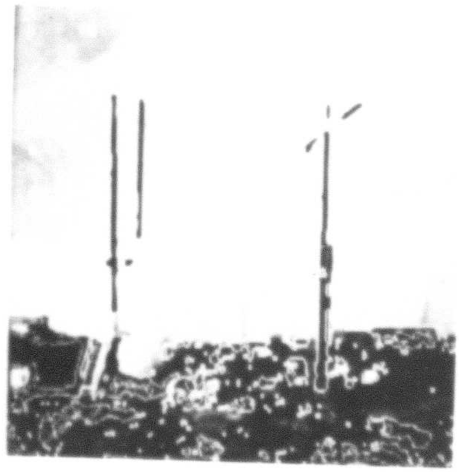


(B) Segmentation

Figure 18 The segmented picture. Local MIN and local MAX used.



(A) Boundary



(B) Segmentation

Figure 19 The segmented picture. Number of local jump and average amp of the jump used. Threshold = 30

Figure 20 The original picture.
Size: 256 x 256

Figure 21 Boundary. Mean and standard deviation used.

Figure 22 Boundary. Local MIN and local MAX used.

Figure 23 The original picture
Size: 256 x 256

Figure 24 Boundary. Mean and standard deviation used.

Figure 25 Boundary. Local MIN and local MAX used.

TEXTURE EDGE DETECTION USING MAX-MIN DESCRIPTORS

S. G. Carlton and O. R. Mitchell

I. Introduction

This report is the sequel to the Nov. 75 - Jan. 76 ARPA Interim Report Article "Texture Edge and Classification Using Max-Min Descriptors". Since that time research concentration in this area has been on improvement of the window averaging technique used previously.

Application results of the min-max descriptors and the improved window averaging are also discussed.

II. Window Averaging (Previous)

Previously, the window averaging used in conjunction with the min-max descriptors involved a variable window size, dependent on picture context, and required two separate processing runs which provided averaging results in both the horizontal and vertical directions. In the horizontal case, the averaging technique compared the total extrema within a window to the right of each pixel point with the total in a like window to the left of each points. Each point is processed individually in this way and replaced by the average calculated as

$$A_H = \frac{R_{tot} - L_{tot}}{R_{tot} + L_{tot} + K} \qquad (1)$$

This average calculation gives maximum values at pixels located on the boundary between texture regions. This same average was then computed using like windows located above and below each pixel. The two averages were then combined and resulting maximum were marked as texture edges.

III. Window Averaging (Now)

The window averaging technique currently in use with the min-max descriptors still uses variable window size based on picture context, but now the horizontal and vertical averages are computed simultaneously, providing faster and more accurate results.

In this technique, the window, NxN, is centered over a particular pixel and four quadrants, N/2 x N/2, are formed as shown below:

B	A
C	D

The resulting average calculation used to replace each individual pixel value in the resulting averaged picture is

$$A = \frac{(B+C) - (A+D) + (B+A) - (C+D)}{2N^2} * \text{scale}$$

The use of the absolute divisor based solely on the window size is because the relative divisor in the previous method suffers from a disturbing quality: an off-center edge produced a bigger output then an on-center edge. The absolute divisor does not suffer this problem.

Although this averaging technique works very well, further research into other possibilities is continuing.

IV. Applications

As a test for the max-min texture technique, an image scanned from the North East Test Area was used as input. Figure 1 shows the 512 x 512 black and white image. Figure 2 shows the detected extrema with intensity used to indicate the size extrema detected. Note that the forested area has a large number of extrema of all sizes. A relatively simple forest detector can be created by thresholding the total number of extrema within a 60 x 60 window surrounding each point. The output of this is shown as full white in Fig. 3.

The texture edges are detected using the window averaging described above. Outputs from this program using window sizes of 60 x 60 and 18 x 18 are shown in Figs. 4 and 5, respectively. The local maximums in Fig. 4 are shown superimposed on the original in Fig. 6. These represent the edges between large (at least 30 x 30) texture regions. Using the various size windows, we are developing a hierarchical structure of texture edge detection. We are also developing an edge detector based on combined texture and intensity information.

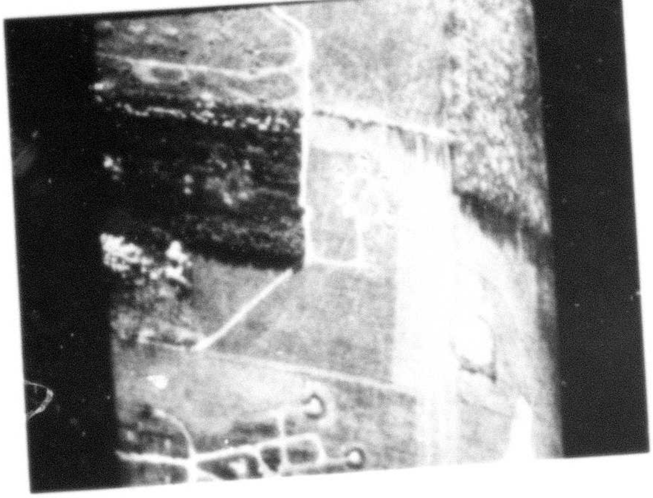


Figure 1 Texture test image.

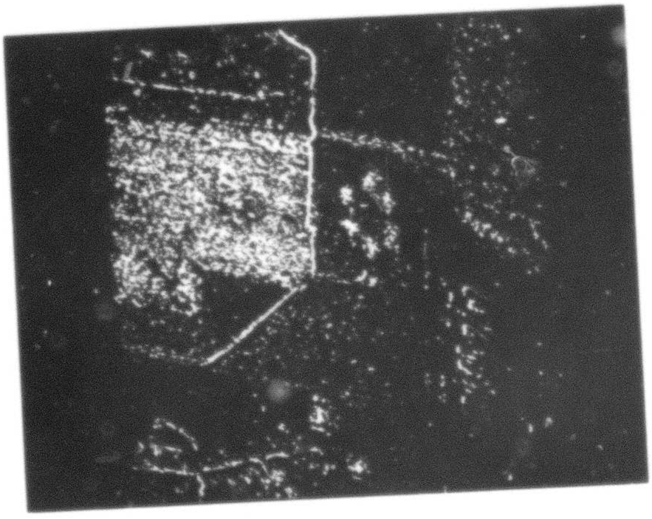


Figure 2 Local extrema in Fig. 1.

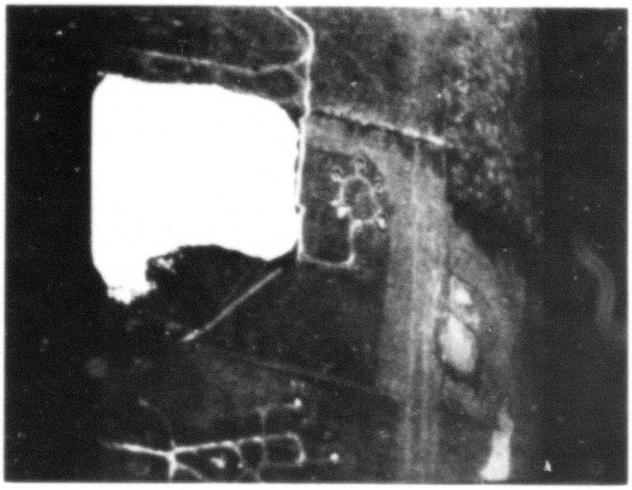


Figure 3 Forest detector output.

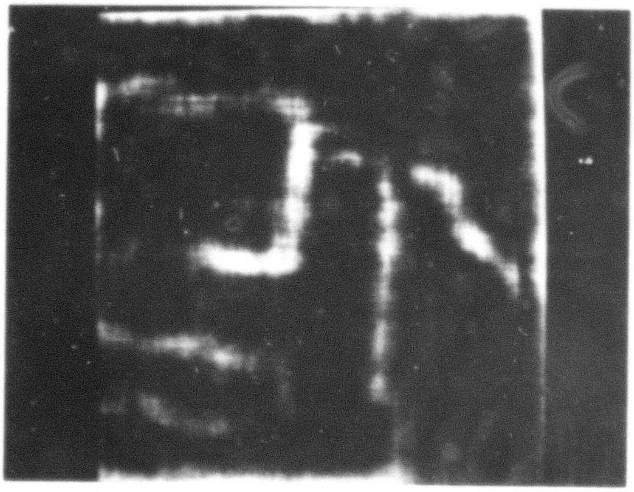


Figure 4 Texture differences using 60 x 60 window.

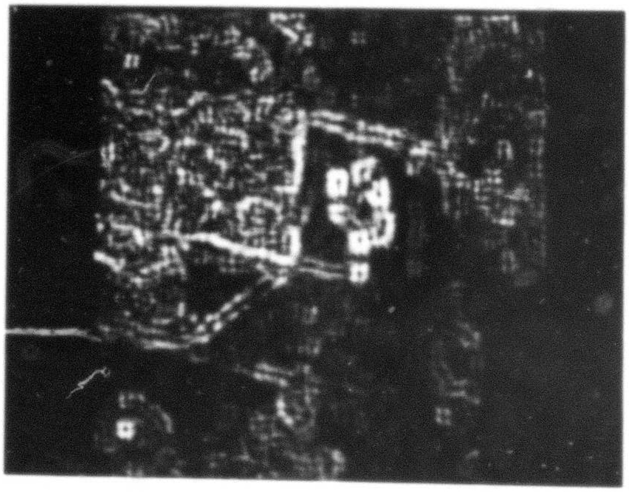


Figure 5 Texture differences using 18 x 18 window.

Figure 6 Local maximums (texture edges) of Fig. 4 superimposed on Fig. 1.

A SYNTAX-DIRECTED METHOD FOR LAND-USE CLASSIFICATION OF LANDSAT IMAGES

K. S. Fu and J. Keng

I. INTRODUCTION

This research is motivated by the need for a method which can fully automate land-use classification, such as, highway, river and bridge recognition, from LANDSAT images. The statistical pattern recognition techniques which have been developed up to now have not shown satisfactory results. For instance, the land-use classification of LANDSAT images has been studied by Todd and Baumgardner using spectral analysis [1]. It has been shown that highways and other concrete areas, such as parking lots, could not be distinguished from each other due to the fact that both have similar spectral characteristics in the spectral analysis. This report introduces a method of spatial analysis for the same purpose of land-use classification without encountering the difficulties mentioned above.

Spatial analysis in picture recognition problems can be treated by syntactic approach [2]. Recently, utilization of syntactic method to describe spatial relationships among different objects was suggested by Fu [3]. Some related research has been done on LANDSAT images. Brayer and Fu [4] recognize a city scene by constructing a hierarchical graph model which contains spatial distributions of all classes in the scene. Web grammars are used to describe spatial relationships between various objects in the scene. Li and Fu [5] started with pointwise statistical classification of LANDSAT images; then applied tree system approach to LANDSAT data interpretation. Bajcsy and Tavakoli [6] designed a computer program from the relational graph viewpoint to recognize objects from satellite pictures.

The research undertaken, here, applies to land-use classification of LANDSAT images such as highway, river and bridge recognition. The suggested approach is a Syntax-Directed Approach [7]. The method based on this approach utilizes a set of tree grammar rules to describe the objects of interest, such as highways and rivers. Accompanied by utilizing semantic information, the application of this method is extended to the problem of bridge recognition.

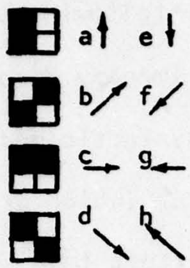
The LANDSAT system (formerly the Earth Resources Technology Satellite "ERTS") consists of three major components; two spacecraft, the remote sensors, and the ground data handling system [8]. The overall system was designed to perform three functions; the acquisition of multispectral images, the collection of data from remotely located sensors, and the production of photographic and digital data. There are four channels; channel 1 (wavelength 0.5-0.6 micrometer), channel 2 (wavelength 0.6-0.7 micrometer), channel 3 (wavelength 0.7-0.8 micrometer) and channel 4 (wavelength 0.8-1.1 micrometer). The first two channels are visible bands and the latter two are infrared bands. LANDSAT images are given in a digitized form by NASA with spatial resolution of one pixel corresponding to 79x56 $(\text{meters})^2$ on the earth.

II. SYNTAX-DIRECTED METHOD

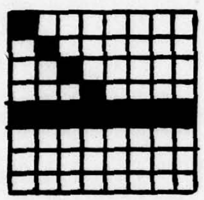
The proposed syntax-directed method involves the following steps: An inference process is applied to a set of training imagery data to infer a set of grammatic rules which in turn formalizes a syntactic model. Based upon this model, a set of most probable window patterns (which are generated by this grammar) is implemented to analyze the test images and to recognize the objects of interest.

2.1 Inference Process

The grammatical inference process is a man-computer interactive system. Based on the knowledge of highway structures, several initial tree grammar rules are written. Then a training area is selected (which in this case is Lafayette, Indiana). The training image is processed by the initial set of grammar rules. An existing highway map is also provided for the purpose of comparison with the processed result. For the highway structures which exist in the map but not in the processed result, the grammar rules to generate those structures are added to the initial set of grammar rules, and the image is processed to test this hypothesis. After several interactive steps the final set of grammar rules is obtained. The primitives for the grammar are chosen as a, b, c, d, e, f, g, and h. These primitives are designed by a 2x2 pixel block.



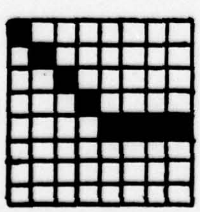
For example, the window area of an intersection of highways is such

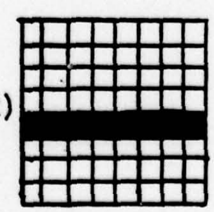


The inferred grammar is

$$S \to \begin{array}{c}\$\\ |\\ A\end{array}\qquad A \to \begin{array}{c} d\\ |\\ A\end{array}\qquad A \to \begin{array}{c} d\\ / \;\; \backslash\\ B \quad C\end{array}\qquad \begin{array}{l} B \to c\\ \\ C \to g\end{array}$$

This inference process continues and at most has 2^{64} possible patterns. The resultant grammar rules are of course too many. We design a simplified tree grammar analyzer using a window operation analyzer. The movement of the window is to shift one column or one row at a time. Then multibranch patterns can also be represented by one-branch grammar rules. For example, the window pattern mentioned above can be analyzed as the following two window patterns.

(1) 

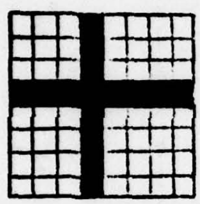
(2) 

The one-branch grammar rules are (A_1, A_2, A_3, A_4 corresponds to the grammar rules presented later).

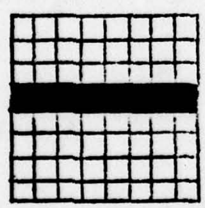
(1) $S \rightarrow \begin{array}{c} \$ \\ | \\ A_3 \end{array}$ $\quad A_3 \rightarrow \begin{array}{c} d \\ | \\ A_3 \end{array}$ $\quad A_3 \rightarrow \begin{array}{c} d \\ | \\ A_4 \end{array}$ $\quad A_4 \rightarrow \begin{array}{c} c \\ | \\ A_4 \end{array}$ $\quad A_4 \rightarrow c$

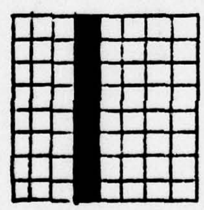
(2) $S \rightarrow \begin{array}{c} \$ \\ | \\ A_1 \end{array}$ $\quad A_1 \rightarrow \begin{array}{c} c \\ | \\ A_1 \end{array}$ $\quad A_1 \rightarrow \begin{array}{c} c \\ | \\ A_2 \end{array}$ $\quad A_2 \rightarrow \begin{array}{c} c \\ | \\ A_2 \end{array}$ $\quad A_2 \rightarrow c$

For a junction of two highways as follows



it can be analyzed by two window patterns that are generated by one-branch grammar rules. The two patterns are





Then the resultant grammar rules can be expressed in terms of only one-branch tree grammar rules as follows[†].

The tree grammar G_t is $G_t = (V, r, P, S)$ where

$$V = \{S, \$, A_1, A_2, A_3, A_4, A_5, A_6, A_7, A_8, A_9, A_{10}, a, b, c, d, e, f, g, h\}$$

$$r(a) = r(b) = r(c) = r(d) = r(e)\ r(f) = r(g) = r(h) = \{1\}$$

[†]Strictly speaking, the simplification essentially reduces the tree grammar to a string grammar. However, the spirit of syntax-directed method is still preserved.

$$V_T = \{\uparrow a,\ \nearrow b,\ \rightarrow c,\ \searrow d,\ \downarrow e,\ \swarrow f,\ \leftarrow g,\ \nwarrow h,\ \$\}$$

and P:

$$S \rightarrow \begin{array}{c}\$\\ |\\ A_1\end{array} \qquad A_1 \rightarrow \begin{array}{c}c\\ |\\ A_1\end{array} \qquad A_1 \rightarrow \begin{array}{c}c\\ |\\ A_2\end{array} \qquad A_2 \rightarrow \begin{array}{c}c\\ |\\ A_2\end{array} \qquad A_2 \rightarrow c$$

$$A_2 \rightarrow \begin{array}{c}d\\ |\\ A_2\end{array} \qquad A_2 \rightarrow d$$

$$A_2 \rightarrow \begin{array}{c}e\\ |\\ A_2\end{array} \qquad A_2 \rightarrow e$$

$$A_2 \rightarrow \begin{array}{c}b\\ |\\ A_2\end{array} \qquad A_2 \rightarrow b$$

$$S \rightarrow \begin{array}{c}\$\\ |\\ A_3\end{array} \qquad A_3 \rightarrow \begin{array}{c}d\\ |\\ A_3\end{array} \qquad A_3 \rightarrow \begin{array}{c}d\\ |\\ A_4\end{array} \qquad A_4 \rightarrow \begin{array}{c}d\\ |\\ A_4\end{array} \qquad A_4 \rightarrow d$$

$$A_4 \rightarrow \begin{array}{c}e\\ |\\ A_4\end{array} \qquad A_4 \rightarrow e$$

$$A_4 \rightarrow \begin{array}{c}f\\ |\\ A_4\end{array} \qquad A_4 \rightarrow f$$

$$A_4 \rightarrow \begin{array}{c}c\\ |\\ A_4\end{array} \qquad A_4 \rightarrow c$$

$$S \rightarrow \begin{array}{c} \$ \\ | \\ A_5 \end{array} \qquad A_5 \rightarrow \begin{array}{c} e \\ | \\ A_5 \end{array} \qquad A_5 \rightarrow \begin{array}{c} e \\ | \\ A_6 \end{array} \qquad A_6 \rightarrow \begin{array}{c} e \\ | \\ A_6 \end{array} \qquad A_6 \rightarrow e$$

$$A_6 \rightarrow \begin{array}{c} f \\ | \\ A_6 \end{array} \qquad A_6 \rightarrow f$$

$$A_6 \rightarrow \begin{array}{c} g \\ | \\ A_6 \end{array} \qquad A_6 \rightarrow g$$

$$A_6 \rightarrow \begin{array}{c} d \\ | \\ A_6 \end{array} \qquad A_6 \rightarrow d$$

$$A_6 \rightarrow \begin{array}{c} c \\ | \\ A_6 \end{array} \qquad A_6 \rightarrow c$$

$$S \rightarrow \begin{array}{c} \$ \\ | \\ A_7 \end{array} \qquad A_7 \rightarrow \begin{array}{c} f \\ | \\ A_7 \end{array} \qquad A_7 \rightarrow \begin{array}{c} f \\ | \\ A_8 \end{array} \qquad A_8 \rightarrow \begin{array}{c} f \\ | \\ A_8 \end{array} \qquad A_8 \rightarrow f$$

$$A_8 \rightarrow \begin{array}{c} g \\ | \\ A_8 \end{array} \qquad A_8 \rightarrow g$$

$$A_8 \rightarrow \begin{array}{c} h \\ | \\ A_8 \end{array} \qquad A_8 \rightarrow h$$

$$A_8 \rightarrow \begin{array}{c} e \\ | \\ A_8 \end{array} \qquad A_8 \rightarrow e$$

$$S \rightarrow \begin{array}{c}\$\\ |\\ A_9\end{array} \qquad A_9 \rightarrow \begin{array}{c}g\\ |\\ A_9\end{array} \qquad A_9 \rightarrow \begin{array}{c}g\\ |\\ A_{10}\end{array} \qquad A_{10} \rightarrow \begin{array}{c}g\\ |\\ A_{10}\end{array} \qquad A_{10} \rightarrow g$$

$$A_{10} \rightarrow \begin{array}{c}a\\ |\\ A_{10}\end{array} \qquad A_{10} \rightarrow a$$

$$A_{10} \rightarrow \begin{array}{c}f\\ |\\ A_{10}\end{array} \qquad A_{10} \rightarrow f$$

$$A_{10} \rightarrow \begin{array}{c}h\\ |\\ A_{10}\end{array} \qquad A_{10} \rightarrow h$$

Measurements from channels 1 and 2 are very sensitive to concrete areas. A training area is set to establish the thresholds of the spectral intensity of concrete areas (more precisely, concrete - like areas) in channels 1 and 2. Then a threshold H is obtained from the sum of two thresholds from these channels. Watery areas exhibit a very low response in the infrared bands. This contrast makes the extraction of watery areas from channels 3 and 4 easier. The same procedure of threshold finding as that for concrete areas is applied here to obtain threshold R for river recognition.

2.2 Syntax-Directed Method

The syntax-directed method consists of two levels, namely, transformation process and tree grammar analysis. The transformation processor transforms the multispectral images into a single binary image. The tree grammar analyzer then analyzes the transformed image based on a tree grammar. Structures which are generated by the tree grammar are accepted; otherwise, they are rejected. The detailed processes are illustrated as follows:

2.2.1 Transformation Process

<u>2.2.1.1 Thresholding Process</u>. First the model of LANDSAT images is defined in the Euclidean n-dimensional space E^n. The number n represents the number of channels to be chosen. A pixel is described by an ordered n-tuple $(X_1, X_2, \ldots, X_n)$. The transformation process works such that if the sum of spectral intensities of the points in the same position in two channels is greater than the sum of the two thresholds from training area (for instance, channel 1, 2, and threshold H for highway recognition, and channel 3, 4, and threshold R for river recognition), the position is set to 1; otherwise, it is set to 0. Thus, multispectral images are transformed to a single binary image. (For river recognition, the one-zero settings are inverse.)

It is true that both visible bands (channel 1 and 2) are sensitive to the concrete spectra. But in real world images, the influence of neighboring objects sometime cause the deformation of the object of interest (such as highway). But when there is only one channel (image) available, the thresholding process can be designed by setting the threshold on one image. Experiments of this case were also conducted and it showed that by using the sum of the spectral intensities of two visible channels (for highway) one obtains a more reliable result than that by just setting a threshold on one channel.

<u>2.2.1.2 Line Smoothing Process</u>. After the thresholding process, a line smoothing technique is applied to remove deformation and reestablish continuity of the lines. For a given center pixel of a 3x3 window,

the operation starts from the left upper corner pixel. If it is one, the column is shifted. If it is zero, the surrounding eight pixels are checked. If there exists at least two "1's" which are not adjacent to each other, then a "1" is set on the center position. The operation continues until reaching the rightmost column of the digitized image. Then the operation is shifted one row down and starts from the left most column with the same process until reaching the last row of the digitized image.

2.2.2 Tree Grammar Analysis

Input: The transformed binary image which is a Q(I,J) memory array.

Output: The syntax-directed analysis result on land use classification.

Algorithm:

Step 1: Set G(M,N) to be an operation window (8x8 in size).

Step 2: Load the array of Q(I,J) where J = 1, 8; I = 1,8 on the operation window G(M,N).

Step 3: Compare the operation window with a set of most probable window patterns (see Fig. 1) which are generated by the tree grammar G_t. If it belongs to that set of patterns, the primitive pattern in that window is accepted, and stored in the resulting memory array R(I,J). If it does not belong to that set of patterns, then go to step 4.

Step 4: Shift one column to the right of Q(I,J) in step 3. Then go to step 3 and continue until reaching the right most column.

2.2.1 Transformation Process

2.2.1.1 Thresholding Process. First the model of LANDSAT images is defined in the Euclidean n-dimensional space E^n. The number n represents the number of channels to be chosen. A pixel is described by an ordered n-tuple $(X_1, X_2, \ldots, X_n)$. The transformation process works such that if the sum of spectral intensities of the points in the same position in two channels is greater than the sum of the two thresholds from training area (for instance, channel 1, 2, and threshold H for highway recognition, and channel 3, 4, and threshold R for river recognition), the position is set to 1; otherwise, it is set to 0. Thus, multispectral images are transformed to a single binary image. (For river recognition, the one-zero settings are inverse.)

It is true that both visible bands (channel 1 and 2) are sensitive to the concrete spectra. But in real world images, the influence of neighboring objects sometime cause the deformation of the object of interest (such as highway). But when there is only one channel (image) available, the thresholding process can be designed by setting the threshold on one image. Experiments of this case were also conducted and it showed that by using the sum of the spectral intensities of two visible channels (for highway) one obtains a more reliable result than that by just setting a threshold on one channel.

2.2.1.2 Line Smoothing Process. After the thresholding process, a line smoothing technique is applied to remove deformation and reestablish continuity of the lines. For a given center pixel of a 3x3 window,

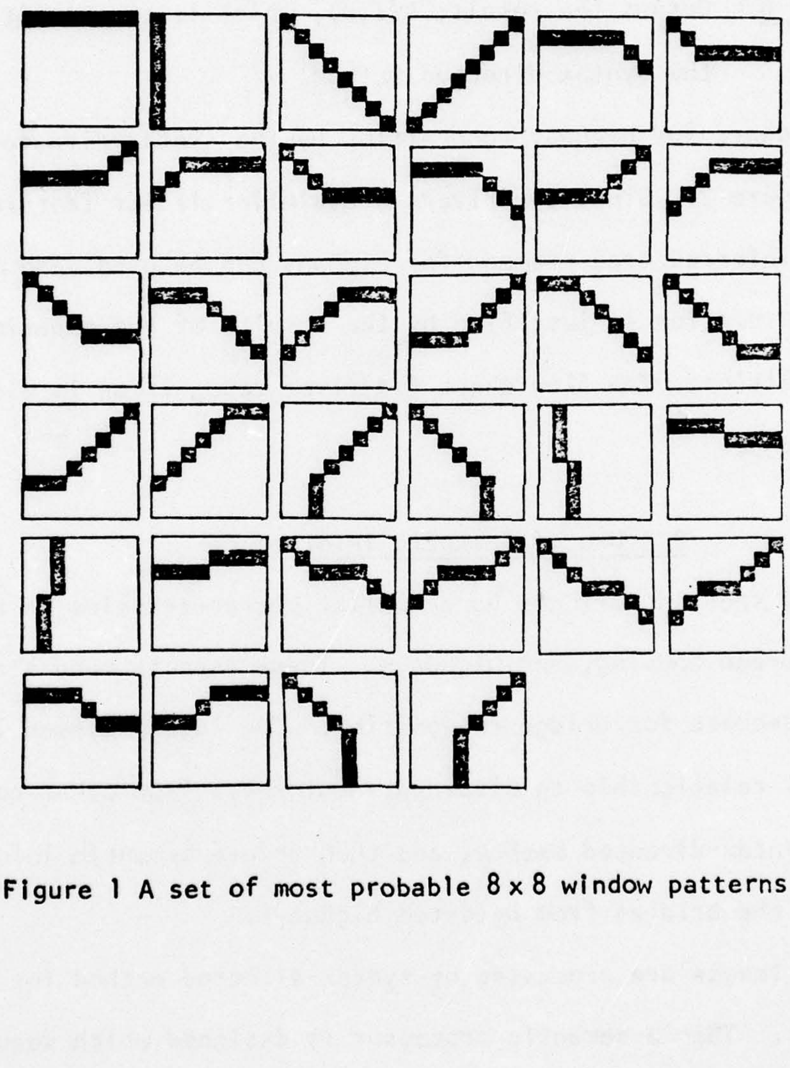


Figure 1 A set of most probable 8×8 window patterns

Step 5 : Shift one row downward in step 3; go to step 3, until reaching the last row of the digitized image. Syntactically correct structures are recognized and stored in the resultant memory array R(I,J).

Step 6 : Output the result, R(I,J), which is the result of the syntax-directed method.

The flow chart for highway recognition by the syntax-directed method is given in Figure 2. Since the rivers are similar linear features to highways, the inferred tree grammar for highway can be used also for river. This assumption is justified by the results of the experiments on river recognition. The flow chart for river recognition is also provided in Figure 3.

2.3 Use of Semantic Information

Spectrally speaking bridges have similar characteristics to concrete parking lots, urban housing, and highways. These aspects make statistical techniques inadequate for bridge recognition. The idea proposed is to use the spatial relationship to distinguish highways from other concrete areas by the syntax-directed method, and then to use semantic information to distinguish the bridges from detected highways.

First the images are processed by syntax-directed method for highways and rivers. Then a semantic processor is designed which sequentially processes semantic rules as follows:

(i) Bridge pixels are highway pixels overlaying water areas (river, lake, or gulf).

(ii) Bridge pixels never exist singularly in the continent.

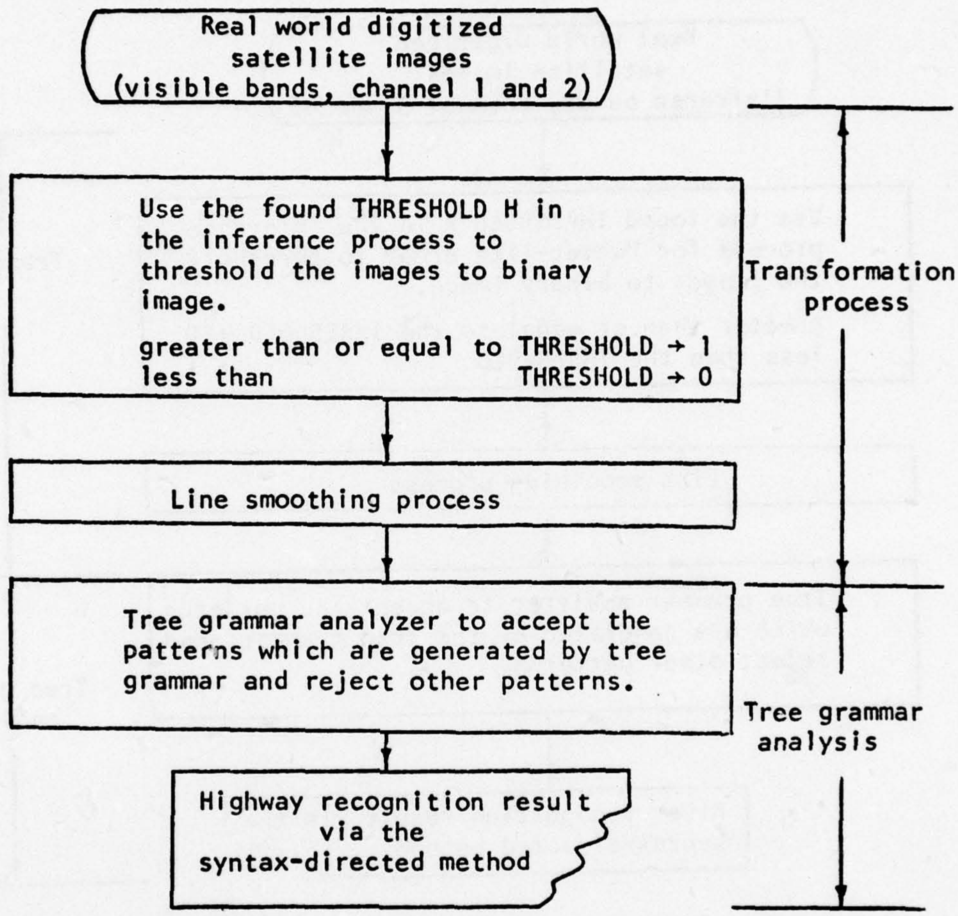


Figure 2 The flow chart of highway recognition via the syntax-directed method

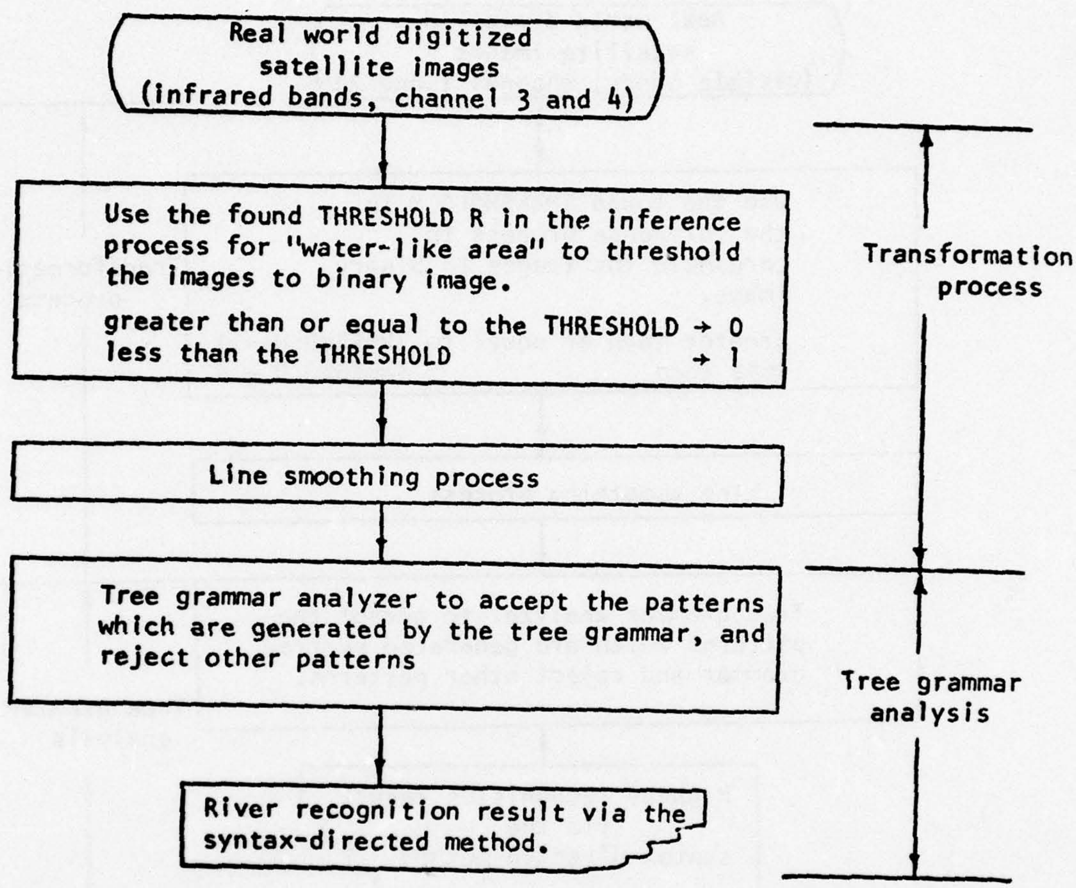


Figure 3 The flow chart of river recognition via the syntax-directed method.

(iii) Both ends of the bridge are always connected to the highways.

A flow chart of the bridge recognition is in Figure 4. The length of the recognized bridge can also be calculated by the following algorithm.

Algorithm of calculation of bridge length:

Input: recognized bridge result.

Output: the calculated length of bridge.

Algorithm:

(i) Calculate the number of horizontal rows which have at least one bridge pixel. The value is a.

(ii) Calculate the number of vertical columns which have at least one bridge pixel. The value is b.

(iii) If a equals one, the length of the bridge c is $b \times 56$ meters.

If b equals one, the length of the bridge c is $a \times 79$ meters.

Otherwise go to (iv)

(iv) The length of the bridge $c = \sqrt{(a\times79)^2 + (b\times56)^2}$.

The idea for this algorithm is to calculate the longest side of a right triangle, and every pixel in LANDSAT images is about 79 meters in vertical length and 56 meters in horizontal length on the earth. Step (iii) are the cases when bridge is right on the horizontal row or vertical column. The coordinate for the locations of the recognized bridges are also located by recording recognized bridge pixels.

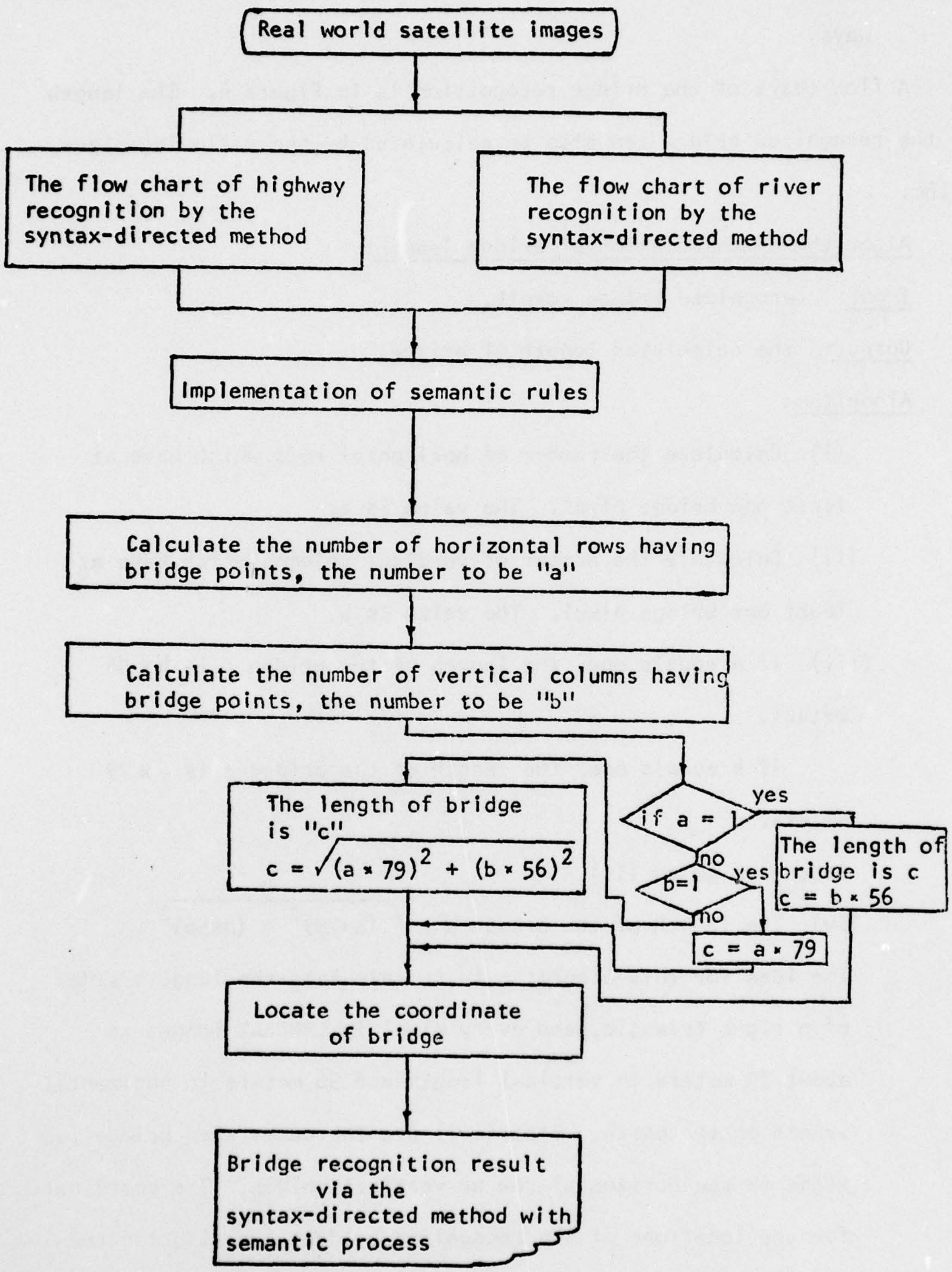


Figure 4 The flow chart of bridge detection via the syntax-directed method with semantic process

III. EXPERIMENTAL RESULTS

The syntax-directed method has been implemented by FORTRAN programming on the IBM 360/67 computer at the Laboratory for Applications of Remotely Sensing (LARS). The experiments have been conducted on different LANDSAT images. Only one training data set (from Lafayette area) was used for all the experiments.

3.1 Highway Recognition

Fig. 5(a) is a LANDSAT image over the Indianapolis, Indiana area. Fig. 5(b) is the intermediate output after line smoothing process in the transformation process. The highway recognition result, by the syntax-directed method, is shown in Fig. 5(c). The area is a 96x96 image which shows the junction of interstate highway 65 (northwest to southeast direction) and highway 465 (north to south direction) in the left upper part of Fig. 5(d). The experimental result shows that the syntax-directed method is rather successful.

3.2 River Recognition

For the purpose of showing that this method works also for rivers, a terrian area northeast of San Francisco, California was processed by the syntax-directed method for river recognition. The LANDSAT image is Fig. 6(a). The river recognition result by the syntax-directed method, Fig. 6(b), shows that it successfully recognizes a winding river in that image. The size of the image is also 96x96. The topographic map for the same area is shown in Fig. 6(c).

Figure 5(a) Satellite image of northwest part of Indianapolis, Indiana

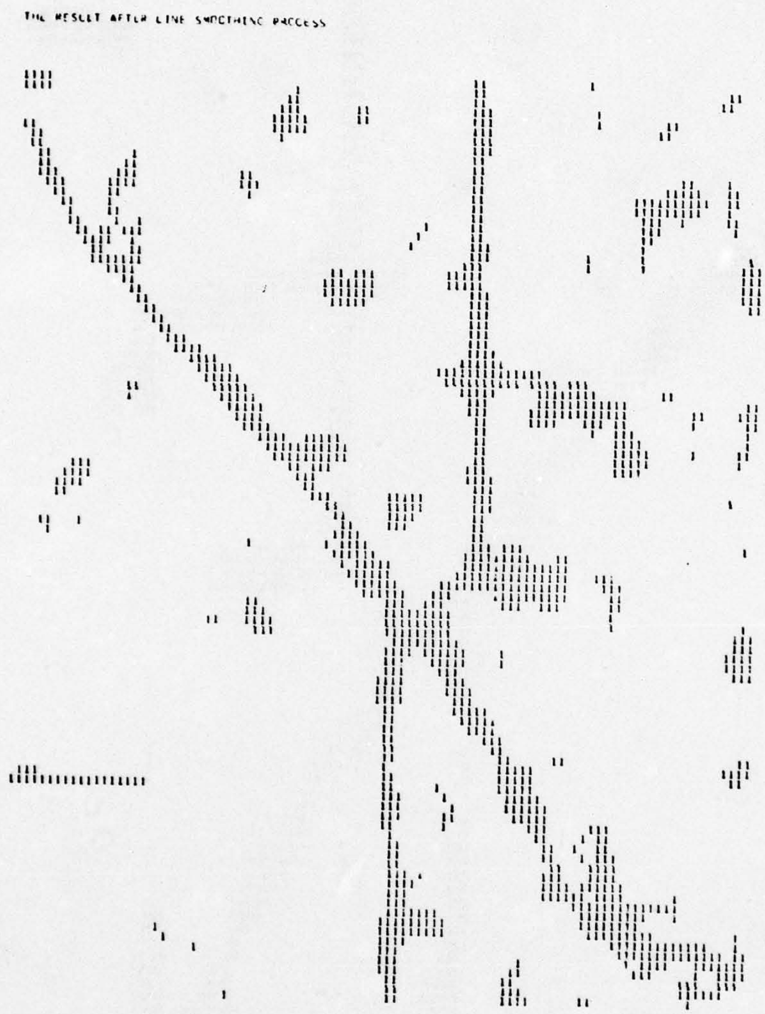


Figure 5(b) Intermidiate output after line smoothing process on Figure 5(a)

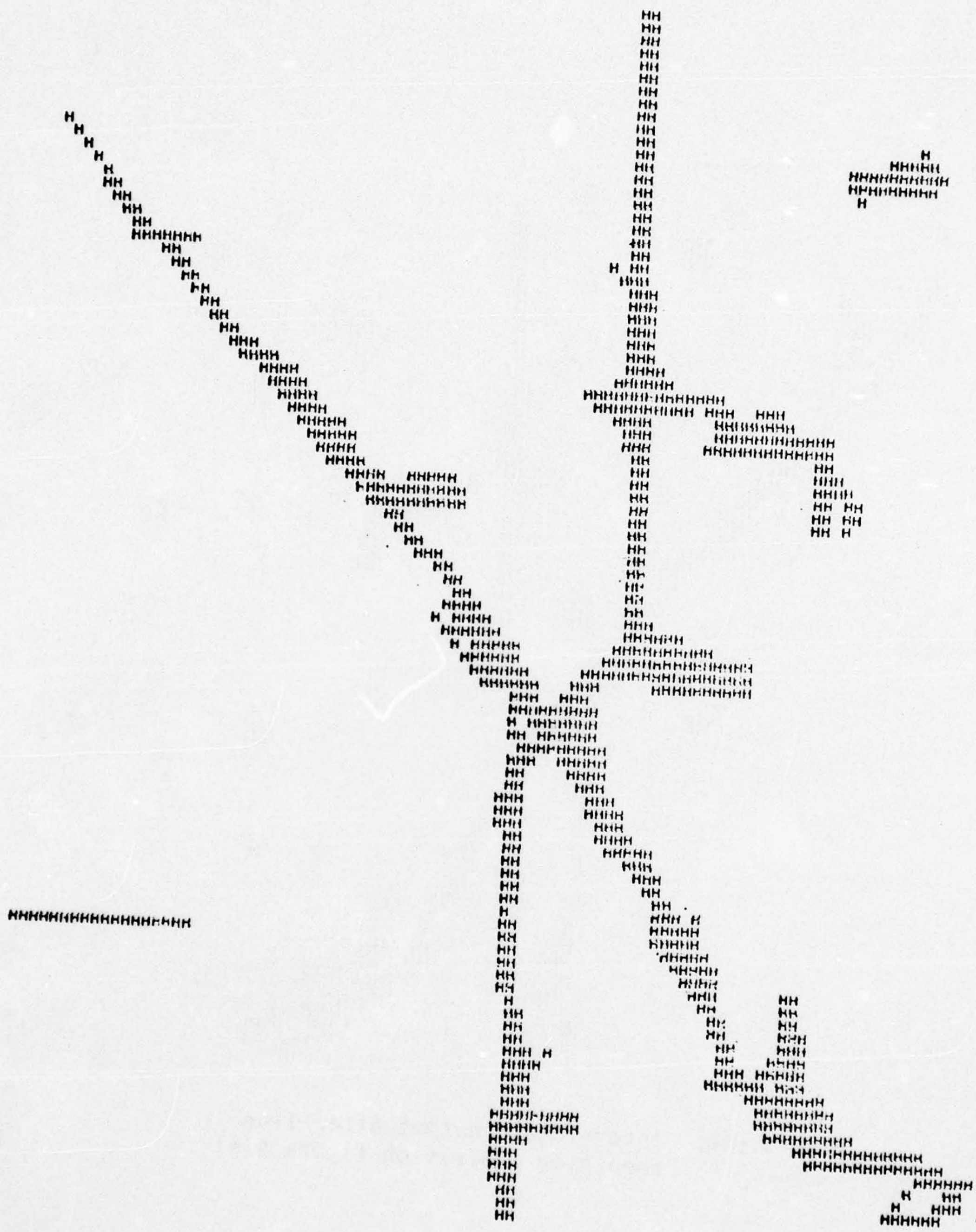


Figure 5(c) Highway recognition result of Figure 5(a) by syntax-directed method

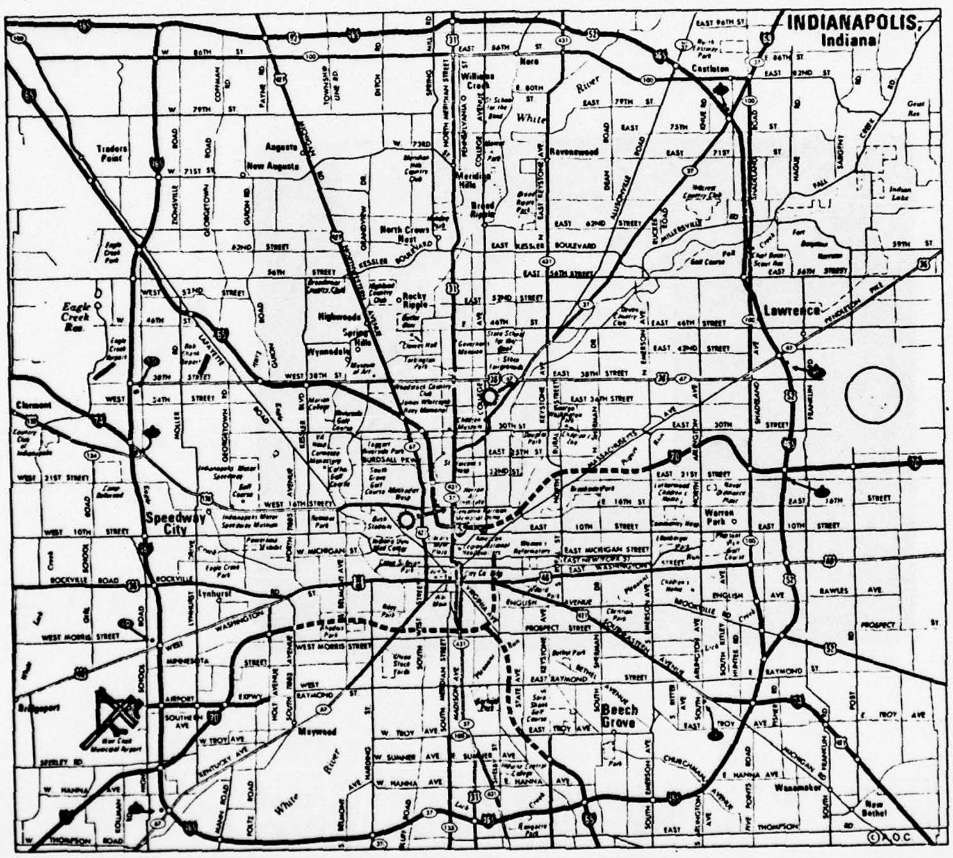


Figure 5(d) City map of Indianapolis, Indiana

Figure 6(a) Satellite image of north part of San Francisco Bay area, California

Figure 6(b) River recognition result of Figure 6(a)
by syntax-directed method

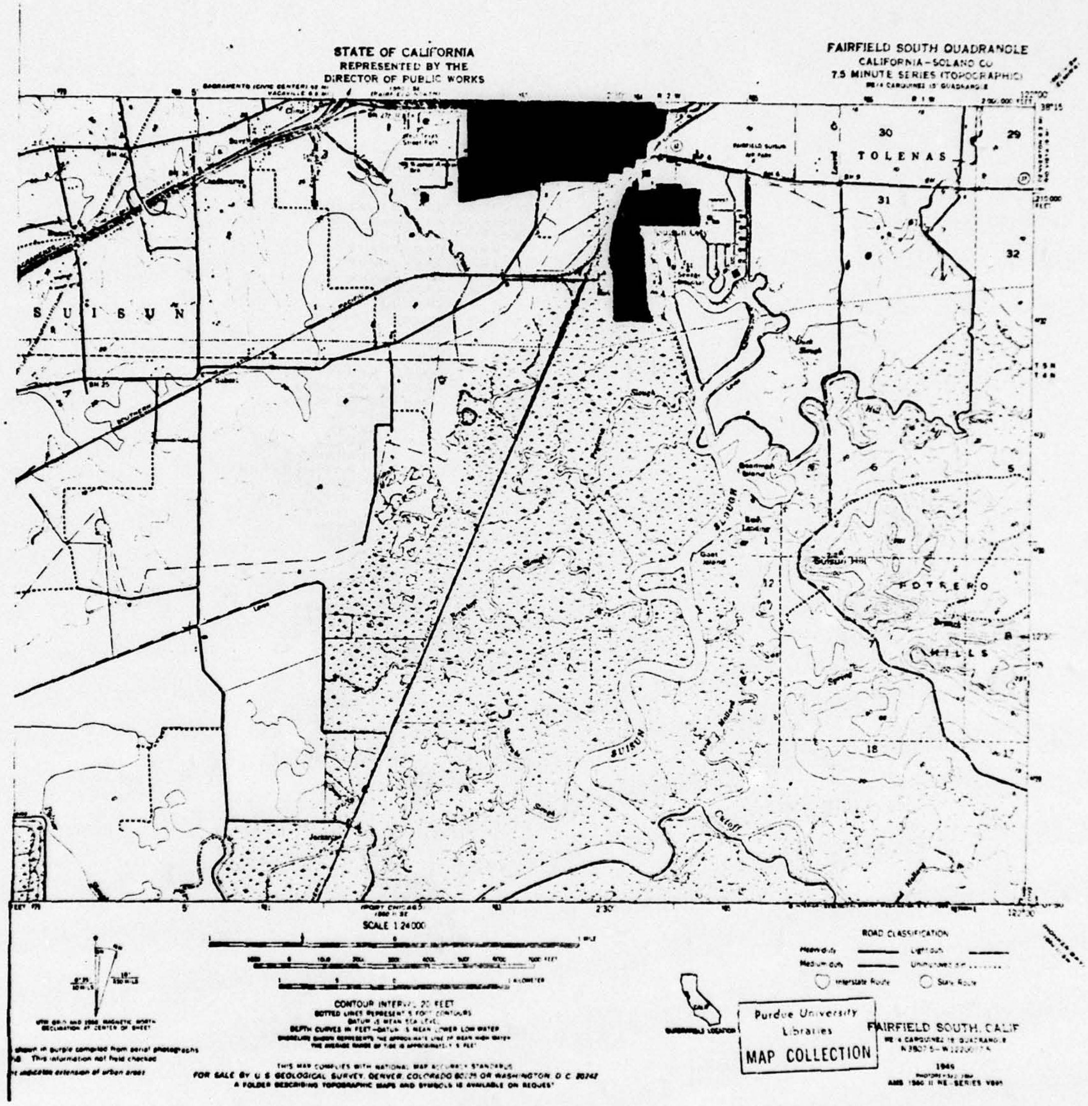


Figure 6(c) Topographic map of the same area of Figure 6(a)

3.3 Bridge Recognition

Fig. 5(a) is a satellite image over Indianapolis, Indiana. Fig. 7(a) is a topographic map of its lower part. It shows in the left part of the map, that there is a bridge over the Eagle Creek Reservoir. The bridge recognition result by the syntax-directed method given in Fig. 7(b) shows that the birdge is successfully recognized and the length is calculated to be 672 meters. In the left lower part of Fig. 7(a) the scale of the map is provided. The length shown in map is about the same as that found by our method. The coordinate of the bridge can also be automatically located by this method.

A third experiment is on the Lafayette area of which the LANDSAT image is shown in Fig. 8(a). Fig. 8(b) is a city map segment of the Lafayette area which shows a small bridge on Highway I-65 over the Wabash River. This LANDSAT image has been processed by the syntax-directed method for bridge recognition and the result is shown in Fig. 8(c). The bridge is recognized in the right lower part of the image and its length is calculated to be 454.1 meters. The coordinates of location of the bridge is also given in Fig. 8(c).

This information extraction (bridge length and coordinate) not only contributes to the understanding of images and may also aid the automatic guidance missile system in locating accurately the objects of interest.

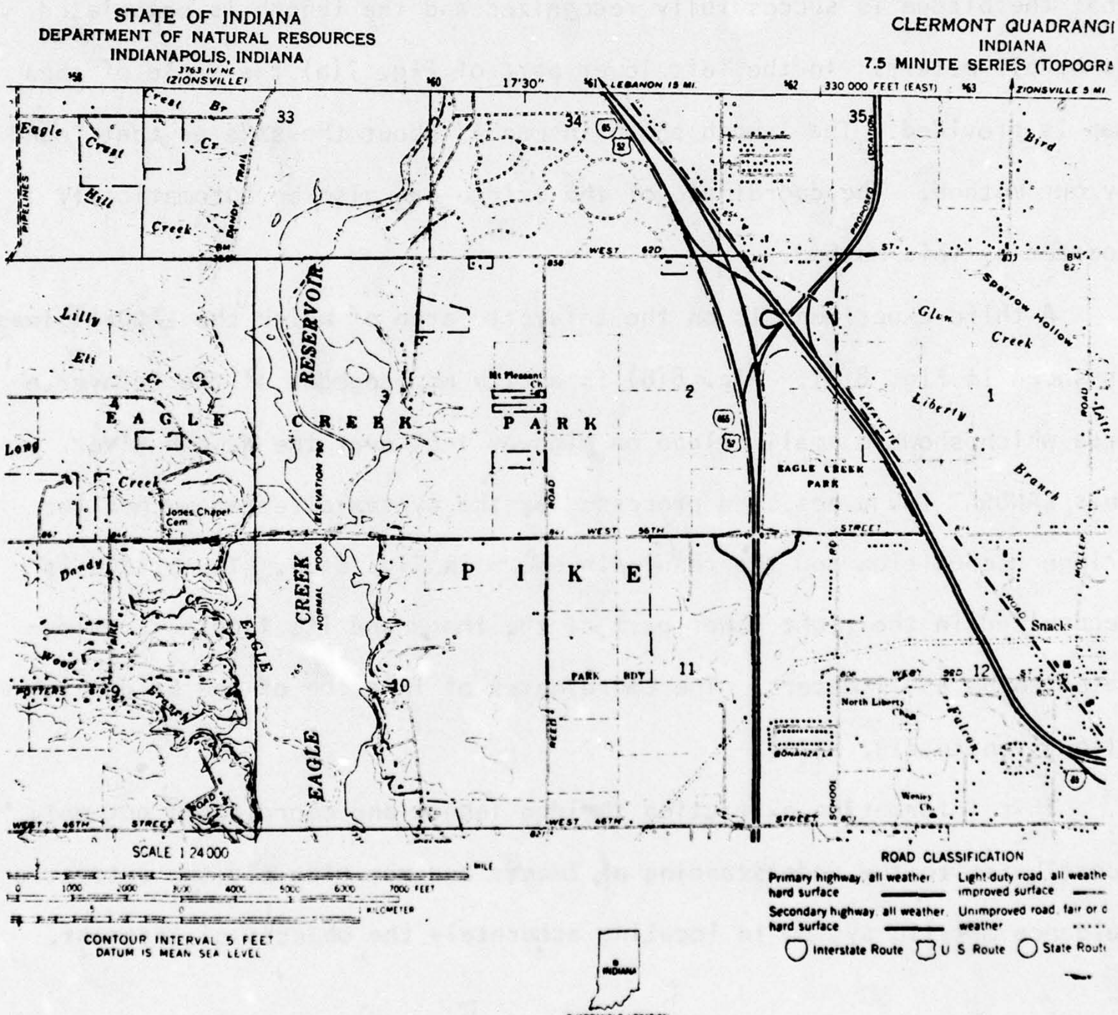
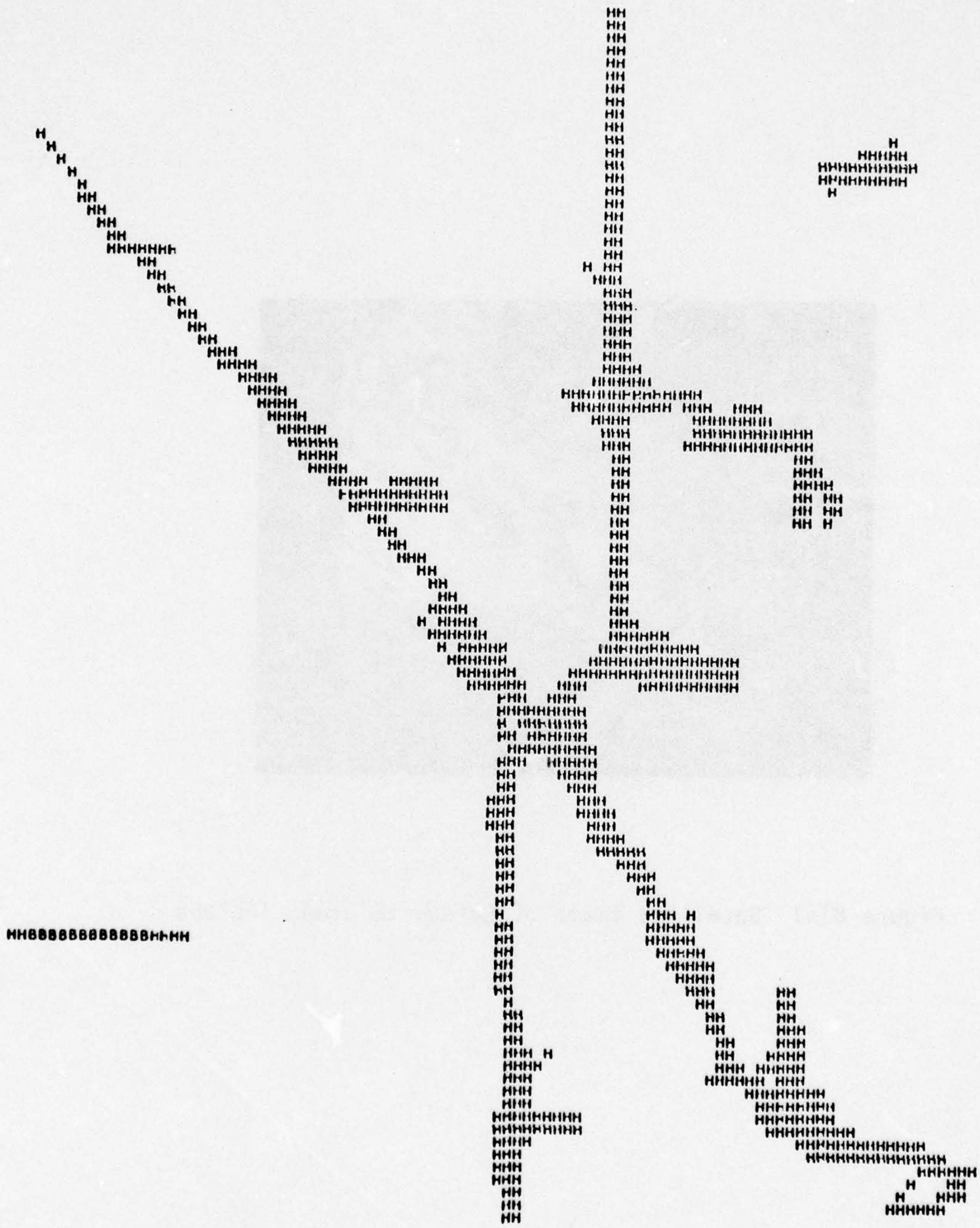


Figure 7(a) Topographic map of lower part of Figure 5(a)



THE CALCULATED LENGTH OF THE DETECTED BRIDGE FROM SATELLITE PICTURE = 672.0 METERS

Figure 7(b) Bridge recognition of satellite image Figure 5(a) by syntax-directed method

Figure 8(a) Satellite image of Lafayette area, Indiana

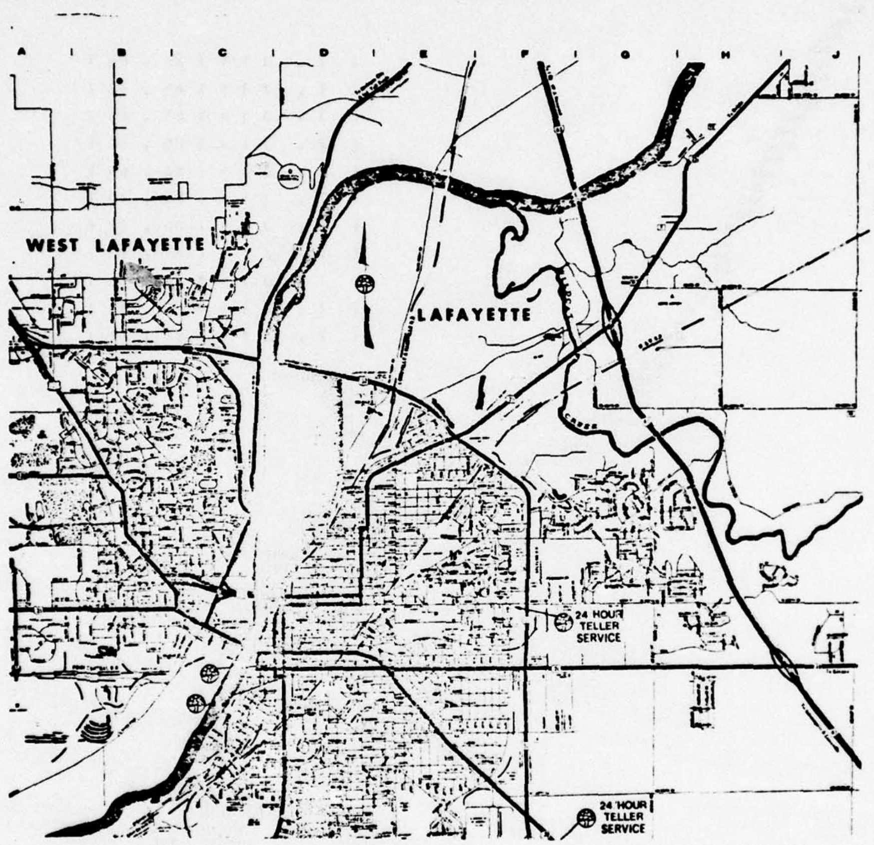
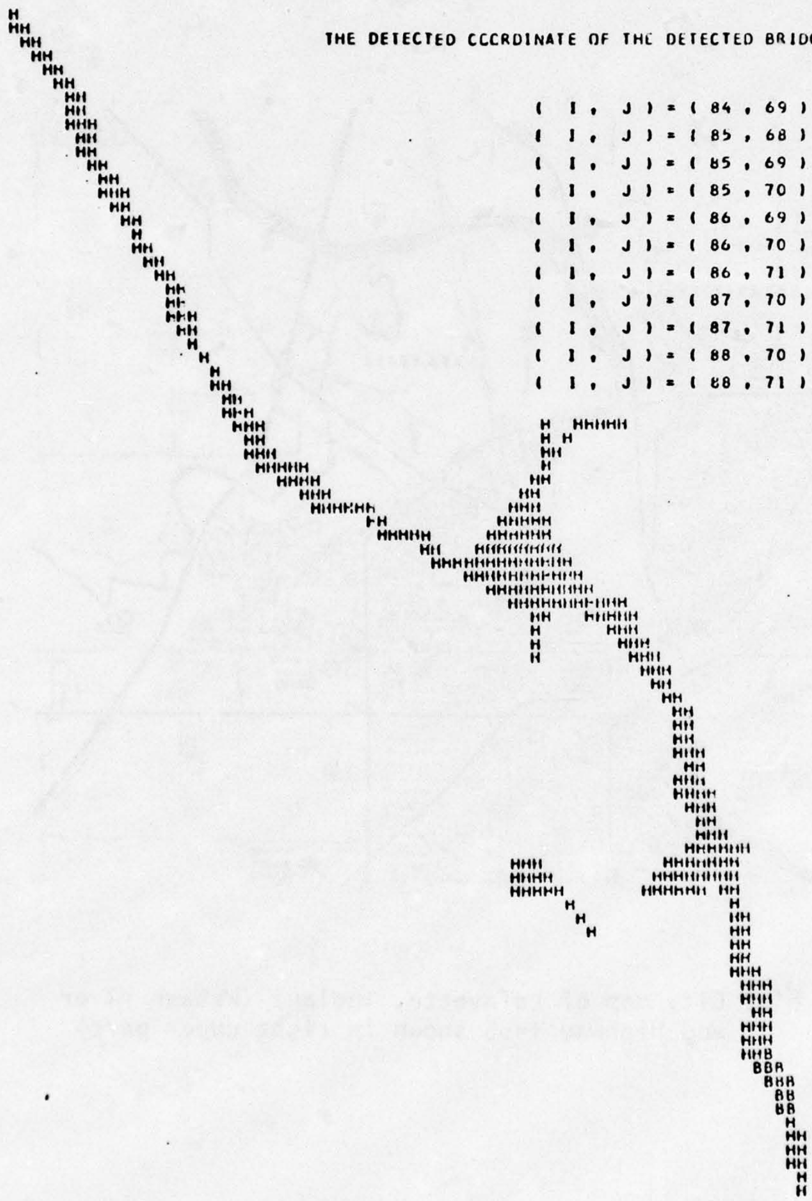


Figure 8(b) City map of Lafayette, Indiana (Wabash river and Highway I-65 shown in right upper part)

THE DETECTED CCCRDINATE OF THE DETECTED BRIDGE FROM SATELLITE PICTURE

(I , J) = (84 , 69)
(I , J) = (85 , 68)
(I , J) = (85 , 69)
(I , J) = (85 , 70)
(I , J) = (86 , 69)
(I , J) = (86 , 70)
(I , J) = (86 , 71)
(I , J) = (87 , 70)
(I , J) = (87 , 71)
(I , J) = (88 , 70)
(I , J) = (88 , 71)



THE CALCULATED LENGTH OF THE DETECTED BRIDGE FROM SATELLITE PICTURE = 454.1 METERS

Figure 8(c) Bridge recognition result of satellite image Figure 8(a) by syntax-directed method

IV. CONCLUSIONS AND REMARKS

The syntax-directed method is implemented by the "LOGICAL programming technique" on binary images. So the software manipulates most operations on the machine logic level. Comparative studies are also carried out with two different implementations. One uses the logical programming and the other does not. The one with logical programming saves 30% of the CPU time comparing with the other. Computer processing time for the syntax-directed method is rather fast compared with the previous related work. For a 96x96 images, the proposed method takes only approximately 26 seconds to detect and recognize highways. It takes approximately a total of 42 seconds to recognize highways, rivers and bridges.

Concerning computer memory space, there is another advantage of LOGICAL programming in that every transformed pixel takes only one byte for storage. Usually each pixel takes 8 bytes (real number) or 4 bytes for storage (integer). Use of the logical programming saves memory space approximately 75% comparing with the one using four bytes storage for each pixel.

The proposed syntax-directed method for land-use classification has the advantages of fast processing time and rather accurate results. It can be easily extended to image segmentation problems.

REFERENCES

1. W. J. Todd and M. F. Baumgardner, "Land-use classification of Marion County, Indiana, by spectral analysis and digitized satellite data," LARS Information Note 101673, LARS, Purdue University, 1973.

2. K. S. Fu, Syntactic Methods in Pattern Recognition, Academic Press, New York, 1974.

3. K.S. Fu, "Pattern recognition in remotely sensing of the earth's resources," IEEE Trans. on Geoscience Electronics, vol. GE-14, pp. 10-18, January 1976.

4. J. M. Brayer and K. S. Fu, "Web grammar and its application to pattern recognition," Ph.D. Thesis, School of Electrical Engineering, Purdue University, West Lafayette, Indiana, 1975.

5. R. Y. Li and K. S. Fu, "Tree system approach for LANDSAT data interpretation," Proceedings of Symposium of Machine Processing of Remotely Sensed Data, Purdue University, West Lafayette, Indiana, pp. 2A-10-pp. 2A-17, 1976.

6. R. Bajcsy and M. Tavakoli, "A computer recognition of bridges, islands, rivers and lakes from satellite pictures," Proceedings of Symposium of Machine Processing of Remotely Sensed Data, Purdue University, West Lafayette, Indiana, pp. 2A-54-pp. 2A-68, October 1973.

7. K. S. Fu and P. H. Swain, "On Syntactic Pattern Recognition" in Software Engineering, vol. 2, ed. by J. T. Tou, Academic Press, 1970.

8. J. J. Horan, "LANDSAT: Multispectral Eye in the Sky," IEEE Spectrum, vol. 13, No. 3, pp. 56-62, March 1976.

THE USE OF CONTEXTUAL INFORMATION IN STATISTICAL CLASSIFICATION

K.S. Fu, P.H. Swain, T.S. Yu, and W. Pfaff

I. Introduction

In previous reports we have introduced compound decision theory [1] as a means to use contextual information in the classification of remote sensing data. The compound decision rule is to choose a_k, for each individual cell k, which minimizes

$$\sum_{\theta_k=1}^{r} L(\theta_k, a_k)\, P(\underset{\sim}{X}_n | \theta_k)\, G(\theta_k) \tag{1}$$

Note that $\underset{\sim}{X}_n$ is the set of pattern vectors for all cells in the imagery frame. The compound decision uses all the measurements from the image frame to estimate the state of nature θ_k. The cells in the data frame are classified simultaneously even though the decision rule is defined for each cell.

Suppose we reduce the data frame to a small window size, say, the 3x3 square window (Fig. 1), the decision rule follows from (1) and is to choose a_k to minimize

$$\sum_{\text{all } \theta_k} L(\theta_k, a_k)\, P(\underset{\sim}{x}_1, \underset{\sim}{x}_2, \ldots \underset{\sim}{x}_9 | \theta_k)\;\; G(\theta_k) \tag{2}$$

If we classify every window from a finite set of possible occurrences of windows, the classification of the center cell, namely cell 1 in Fig. 1, is then a result of the (spectral) properties of its neighbors as well as its own. Thus, the "context" defined in the scene contributes to the classification.

II. Preliminary Solution

Immediately we see that the calculation of the joint probability for the cells in the window is necessary. Probability measure defined on multidimensional space is in the theory of random field. We do not intend to go

8	3	7
4	1	2
9	5	6

Figure 1 Neighbors of cell 1

into the theoretical development but rather we give the research result.

Suppose we assume the probability distribution belongs to an exponential family. And suppose that the dependence is from cliques* containing no more than two cells. The joint normal density function for any n cells in our window is [2]

$$P(x_1 .. x_n) = 2\pi\sigma^2)^{-\frac{1}{2}n} |B|^{\frac{1}{2}} \exp\{-\frac{1}{2}\sigma^{-2}(\underset{\sim}{X}-\underset{\sim}{U})^T \underset{\sim}{B} (\underset{\sim}{X}-\underset{\sim}{U})\} \qquad (3)$$

where $\underset{\sim}{U}$ is nxl vector of arbitrary means, μ_i, and $\underset{\sim}{B}$ is the nxn matrix whose diagonal elements are unity and whose off-diagonal elements are $-\beta_{ij}$. We require B to be positive definite.

One important feature about this formulation is that the window is not restricted to any particular shape. As long as we know the neighboring cells to be used to provide context relationship, it is possible to construct the joint function. The unknown parameters in (3) are $\underset{\sim}{B}$ matrix and σ_i^2. In a subsequent report we shall show estimation of unknown parameters for a particular window (nearest neighbor scheme) and present some results.

---

*Clique: Any set of cells which either consists of a single cell or else in which every cell is a neighbor of every other cell in the set

III. Data Simulation

The algorithms we are investigating ordinarily require various assumptions concerning the statistical characteristics of the data. For example, it is often assumed that the data is class-conditionally distributed multivariate normal and, when classification is involved, that the data used to train the classifier is truly representative of the classes of interest. It is further assumed that all classes present in the scene are known and represented. Also, it is usually assumed that each pixel consists entirely of one and only one of the classes.

In practice these assumptions are never exactly satisfied. Thus, when we evaluate the results of applying our algorithms to any real data set, we are faced with uncertainty as to whether the errors we observe are due to shortcomings of the algorithms per se or to some assumptions we made about the data in order to apply the algorithms. We can not then determine effectively where the strengths and weaknesses of our approach lie.

The purpose of this data simulation activity, therefore, is to make available data sets with controlled characteristics. In particular, to begin with we shall generate simulated LANDSAT multispectral data. The inputs to the simulation will be high-quality classifications of selected ground scenes together with the estimated mean vectors and covariance matrices of the classes in the scene. In the output, the multispectral data values will be regenerated so that the classes are in fact multivariate normally distributed and there are no mixture pixels. Since we will have started with an accurate classification, the spatial characteristics of the data will closely match those of the actual ground scene, but the multispectral characteristics will be controlled to meet the multivariate normal assumption.

This data will be used specifically to test our methods for extracting contextual information (which is primarily of a spatial orientation) for use in classification. In the next report we shall detail the methods used to accomplish the simulation.

References

[1] K. Abend, "Compound decision procedure for unknown distribution and for dependent state of nature," in Pattern Recognition, L. Kanal, Ed. Washington, D.C.: Thompson, pp. 207-249, 1968.

[2] J. E. Besag, "Spatial interaction and the statistical analysis of lattice systems," Journal Royal, Stat. Soc. B 36, pp. 192-236, 1974.

TWO-DIMENSIONAL COMPLEX CEPSTRUM

B. O'Connor and T. S. Huang

1. Introduction

This report will present the details on the theory and implementation of two-dimensional complex cepstral analysis referred to in the last report. Possible uses of this algorithm include tests for stability of two-dimensional digital recursive filters and enhancement of images blurred by special point-spread functions. Cannon [1] has used the cepstral signature of a blurred picture to determine the extent and the type of degradation. However, he allowed only three kinds of blurs; linear motion, defocus, and atmospheric. These degradations have real point-spread functions and their zero crossings (phase information) characterize them completely. In fact, since his method used the cepstrum, as opposed to the complex cepstrum, zero crossings give the only phase information about the blur and hence this technique cannot be employed to deblur more complicated degradations. The algorithm to be reported here will allow the calculation of the complex cepstrum of a picture. The advantage of the complex cepstrum is that it uses both the magnitude and phase of the input picture and hence allows a complete picture description in the cepstral domain. Hopefully, this description will give cepstral signatures to more complicated blurring functions.

The complex cepstrum can be employed to test the stability of general two-dimensional recursive filters. The complex cepstrum of a state two-dimensional recursive filter is non-zero only in a certain distinguished regions of the cepstral plane. Ekstrom and Woods have obtained similar results employing the cepstrum (not the complex cepstrum).

II. 2-D Complex Cepstrum Theory

Let $x(m,n)$ be a $M \times N$ sequence of real numbers. The complex cepstrum of $x(m,n)$ called $cx(m,n)$ is defined as the inverse z-transform of the complex logarithm of the z-transform of $x(m,n)$. Let $X(z_1,z_2)$ be the z-transform of $x(m,n)$ and $\hat{X}(z_1,z_2) = \log X(z_1,z_2)$. $\hat{X}(z_1,z_2)$ must be a valid z-transform for $cx(m,n)$ to exist. In order for $cx(m,n)$ to be uniquely defined, a region of convergence must be chosen for $\hat{X}(z_1,z_2)$. Assume $x(m,n)$ and $cx(m,n)$ are real, stable sequences so that the regions of convergence of both $X(z_1,z_2)$ and $\hat{X}(z_1,z_2)$ include the unit polydisc and hence, their Fourier Transforms exist.

$$X(e^{j\omega_1}, e^{j\omega_2}) = X_R(e^{j\omega_1}, e^{j\omega_2}) + j\, X_I(e^{j\omega_1}, e^{j\omega_2})$$

$$\hat{X}(e^{j\omega_1}, e^{j\omega_2}) = \hat{X}_R(e^{j\omega_1}, e^{j\omega_2}) + j\, \hat{X}_I(e^{j\omega_1}, e^{j\omega_2})$$

Since $cx(m,n)$ is real, then $\hat{X}_R$ must be an even function of (ω_1,ω_2) and $\hat{X}_I$ must be an odd function of (ω_1,ω_2). More importantly since $\hat{X}$ is analytic then it must be a continuous function of (ω_1,ω_2). Hence, since

$$X(e^{j\omega_1}, e^{j\omega_2}) = |X(e^{j\omega_1}, e^{j\omega_2})| \cdot e^{j\arg[X(e^{j\omega_1}, e^{j\omega_2})]}$$

implies that

$$\hat{X}(e^{j\omega_1}, e^{j\omega_2}) = \log |X(e^{j\omega_1}, e^{j\omega_2})| + j \arg[X(e^{j\omega_1}, e^{j\omega_2})]$$

so

$$\hat{X}_R(e^{j\omega_1}, e^{j\omega_2}) = \log |X(e^{j\omega_1}, e^{j\omega_2})|$$

$$\hat{X}_I(e^{j\omega_1}, e^{j\omega_2}) = \arg[X(e^{j\omega_1}, e^{j\omega_2})]$$

must be continuous functions of (ω_1,ω_2). However, continuity of $\hat{X}_I$ is dependent on the definition of the complex logarithm. Here a difficulty arises since the complex logarithm is not a unique transformation. This difficulty cannot be resolved by using $\mathrm{ARG}[X(e^{j\omega_1}, e^{j\omega_2})]$ because ARG is a discontinuous function.

One approach for obtaining a continuous phase is to integrate its derivative. The resultant phase curve is called the unwrapped phase. Below, the phase derivative is calculated in terms of easily obtained quanties. Starting with

$$\frac{\partial}{\partial z_1} \log[X(z_1,z_2)] = \frac{1}{X(z_1,z_2)} \cdot \frac{\partial X(z_1,z_2)}{\partial z_1}$$

this implies that

$$\frac{\partial}{\partial \omega_1} \hat{X}(e^{j\omega_1}, e^{j\omega_2}) = \frac{\partial X(e^{j\omega_1}, e^{j\omega_2})}{\partial \omega_1} \Big/ X(e^{j\omega_1}, e^{j\omega_2})$$

$$= \frac{\partial}{\partial \omega_1} \hat{X}_R(e^{j\omega_1}, e^{j\omega_2}) + j \frac{\partial}{\partial \omega_1} \hat{X}_I(e^{j\omega_1}, e^{j\omega_2})$$

hence

$$\frac{\partial}{\partial \omega_1} \arg[X(e^{j\omega_1}, e^{j\omega_2})] = \frac{\partial}{\partial \omega_1} \hat{X}_I(e^{j\omega_1}, e^{j\omega_2})$$

$$= \frac{X_R \cdot \frac{\partial}{\partial \omega_1} X_I - X_I \cdot \frac{\partial}{\partial \omega_1} X_R}{X_R^2 + X_I^2}$$

with boundary condition

$$\arg[X(e^{j\omega_1}, e^{j\omega_2})]\Big|_{\substack{\omega_1=0 \\ \omega_2=0}} = 0$$

Note that in some cases it will be necessary to change the signs of all x(m,n) before using this boundary condition. The expression for the phase derivative can be calculated directly from the sequence x(m,n) if we employ the following relation.

$$\frac{\partial}{\partial\omega_1} X(e^{j\omega_1}, e^{j\omega_2}) = \frac{\partial}{\partial\omega_1} X_R(e^{j\omega_1}, e^{j\omega_2}) + j\frac{\partial}{\partial\omega_1} X_I(e^{j\omega_1}, e^{j\omega_2})$$

$$= -j\ FT[m\cdot x(m,n)]$$

A similar formula can be derived for the partial derivative of the phase with respect to ω_2.

An efficient phase unwrapping algorithm has been proposed recently by J. Tribolet [3]. The algorithm uses an adaptive numerical integration scheme that combines the information contained in both the phase derivative and the principal value of the phase. Each phase estimate is formed by numerical integration of the phase derivative using a given step interval. This step interval is adapted until the principal value of the resultant phase estimate does not significantly differ from the known principal value of the phase at that frequency. This method has been extended to two dimensions in our present work. The basic idea of the algorithm in two dimensions is to calculate the integral of the partial derivatives by numerical means using the trapezoidal integration rule. Assuming the unwrapped phase of $(\omega_{01},\omega_{02})$ is known, the estimate of the unwrapped phase at $(\omega_{11},\omega_{02})$ is given by

$$\overline{\arg}[X(e^{j\omega_{11}}, e^{j\omega_{02}})] = \arg[X(e^{j\omega_{01}}, e^{j\omega_{02}})]$$

$$+ \frac{\omega_{11}-\omega_{01}}{2} \cdot \{\frac{\partial}{\partial\omega_1} \arg[X(e^{j\omega_{01}}, e^{j\omega_{02}})$$

$$+ \frac{\partial}{\partial\omega_2} \arg[X(e^{j\omega_{11}}, e^{j\omega_{02}})]\}$$

A similar equation is true in the ω_2 direction. Clearly, this estimate improves as the step interval becomes smaller. The basic idea of this algorithm is to adapt the step size until the result of the numerical integration matches the information given by the principal value of the phase [3].

We shall say that the step size $\Delta\omega = \omega_{11} - \omega_{01}$ leads to a consistent phase estimate of $(\omega_{11}, \omega_{02})$ if

$$|E((\omega_{01}, \omega_{02}) , (\omega_{11}, \omega_{02}))| << \pi$$

Where E measures the difference between the principal values of the phase and its estimate, that is

$$E((\omega_{01}, \omega_{02}) , (\omega_{11}, \omega_{02})) \triangleq \overline{\mathrm{ARG}}[X(e^{j\omega_{11}}, e^{j\omega_{02}})] - \mathrm{ARG}[X(e^{j\omega_{11}}, e^{j\omega_{02}})]$$

Otherwise the step size must be reduced in order to obtain a more reliable estimate of the phase. As soon as a reliable estimate is found the unwrapped phase is defined by

$$\arg[X(e^{j\omega_{11}}, e^{j\omega_{02}})] = \overline{\arg}[X(e^{j\omega_{11}}, e^{j\omega_{02}})] - E((\omega_{01}, \omega_{02}) , (\omega_{11}, \omega_{02}))$$

so it unwraps to

$$\arg[X(e^{j\omega_{11}}, e^{j\omega_{02}})]$$

without error.

If is comforting to know that the existence of cepstra for 2-D rational polynomials has been proved by Dudgeon [4]. In general, if $x(m,n)$ has a rational z-transform then the phase associated with $X(e^{j\omega_1}, e^{j\omega_2})$ has a linear component plus a continuous, odd, periodic component. From this information it can be shown that any 2-D array having a rational z-transform will have a well defined 2-D complex cepstrum if $X(e^{j\omega_1}, e^{j\omega_2}) \neq 0$ for all (ω_1, ω_2), and if the linear phase components have been eliminated by an appropriate shift of the original array. On this issue of linear phase, there is a significant departure from one-dimensional theory in that the linear phase component cannot be completely removed merely by shifting $x(n,m)$ by an appropriate amount [6].

III. Summary

The above two-dimensional phase unwrapping algorithm and complex cepstrum computer programs have been written using the two-dimensional FFT to approximate the Fourier Transforms. These programs have been applied successfully in determining stability of two-dimensional recursive filters. Future work will include extending this program so that the cepstrum of a picture can be calculated on a PDP-11/45.

References

1. T. M. Cannon, "Digital Image Deblurring by Nonlinear Homomorphic Filtering," Ph.D. Thesis, University of Utah, Salt Lake City, Utah, August 1974.

2. M. Ekstrom and J. Woods, "Two-Dimensional Spectral Factorization with Applications in Recursive Digital Filtering," IEEE Trans. ASSP, Vol. 24, pp. 115-127, April 1976.

3. J. Tribolet, "A New Phase Unwrapping Algorithm," submitted to IEEE Trans. ASSP, March 1976.

4. D. Dudgeon, "Two-Dimensional Recursive Filtering," Ph.D. Thesis, MIT, Cambridge, MA, 1974.

5. A. Oppenheim and R. Schafer, Digital Signal Processing, Prentice-Hall, Inc., 1975.

6. A. Filip, "Estimating the Impulse Response of a Linear, Shift-Invariant, Image Degrading System," Ph.D. Thesis, MIT, Cambridge, MA, 1972.

IMAGE RESTORATION: COMPARISON OF THE PROJECTION METHOD WITH SINGULAR VALUE DECOMPOSITION (SVD)

S. P. Berger and T. S. Huang

I. Introduction

This report contains a comparison of two techniques of image restoration. The projection method and the singular value decomposition (SVD) approach were applied to a simple two-dimensional image which was blurred by a linear degradation. White Gaussian noise was added to the degraded image. The two methods were compared for varying amounts of noise.

Before presenting the computer results, it will be helpful to give the basic ideas of the two methods.

II. The Projection Algorithm

The projection method requires that the degradation process be represented in discrete form. A two-dimensional image is represented as a one-dimensional matrix by stacking the rows of pixel values into a column vector. The degradation is then described by

$$g_1 = a_{11} f_1 + a_{12} f_2 + \dots + a_{1n} f_n + n_1$$

$$g_2 = a_{21} f_1 + a_{22} f_2 + \dots + a_{2n} f_n + n_2$$

$$\vdots$$

$$g_n = a_{n1} f_1 + a_{n2} f_2 + \dots + a_{nn} f_n + n_n$$

where $\underline{f}$ is an nx1 matrix (column vector) representing the original image, $\underline{g}$ is the degraded image, $\underline{n}$ is a noise vector, and $\underline{A}$ is the degrading matrix. For the sake of simplicity, we assume that the original image is a square array of NxN elements. So the vector $\underline{f}$ contains $N \cdot N = n$ elements. The matrix equation is $g = \underline{A}\,\underline{f} + \underline{n}$.

Let us neglect the noise vector for the moment. The restoration problem is to obtain $\underline{f}$ from $\underline{g}$, with $\underline{A}$ unknown. The projection method is an iterative procedure for solving for the original image $\underline{f}$.

Each equation represents a hyperplane in an n-dimensional space. The degraded image $\underline{g}$ is a point in this space. The projection method starts at some initial guess (usually $\underline{g}$), and finds the projection onto the first hyperplane. This point is then projected onto the second hyperplane, and so on, until the last projection is onto the nth hyperplane. This then completes one major iteration. A projection is made onto the first hyperplane to start the next major iteration.

If a unique solution exists, the algorithm will yield vectors which converge to it. If no such solution exists, the method still provides useful results. The effect of additive noise tends to increase with more iterations. The decision must then be made as to how many iterations to perform. This is a subjective determination of which iteration has yielded the best restored image.

III. Singular Value Decomposition

The singular value decomposition (SVD) approach is based on a representation of the pseudo-inverse of a matrix. The original image $\underline{f}$ can be estimaged by $\hat{\underline{f}} = \underline{A}^{+} \underline{g}$, where A^{+} is the Moore-Penrose pseudo-inverse. Now A^{+} can be obtained by $\underline{A}^{+} = \sum_{i=1}^{R} \frac{1}{(\lambda_i)^{1/2}} \underline{V}_i \underline{U}_i^T$, where R is the rank of $\underline{A}$, $\underline{U}_i$ and V_i are the eigenvectors of $\underline{A}\,\underline{A}^T$ and $\underline{A}^T\underline{A}$, respectively, and the λ_i are the eignevalues of either (called the singular values of $\underline{A}$). The λ_i are in the order of decreasing magnitudes.

The restoration is then represented as $\hat{\underline{f}} = \sum_{i=1}^{P} \frac{1}{(\lambda_i)^{1/2}} \underline{V}_i \underline{U}_i^T \underline{g}$. The user must decide on the optimal value of P, since the effect of noise will dominate the summation after a certain number of terms. This effect is demonstrated in the restoration equation $\underline{f} = \underline{A}^+ \underline{g} = \underline{A}^+[\underline{A}\,\underline{g} + \underline{n}] = \underline{A}^+ \underline{A}\,\underline{g} + \underline{A}^+ \underline{n}$. The first term, $\underline{A}^+ \underline{A}\,\underline{g}$, is the minimum-norm least-square estimate without noise, and the second is the noise term. The larger the value of P, the closer the first erms is to the original image. The noise term, however, increases as $\frac{1}{\lambda_i^{1/2}}$ with P.

IV Computer Results

Both of these methods were applied to a problem given in Huang and Narendra [1]. An 8x8-pixel image of the number "5" was used in the test (Fig. 1). The blank area has a value 0, and the dark region has the value 7. This image was degraded by replacing each point with the average of the 3x3 array containing the point in the center. The degraded image is given in Fig. 2.

This degradation results in a sparse matrix, $\underline{A}$, (64x64) containing only the values 0 and 1/9. On the border, this choice of $\underline{a}$ reflects the assumption that the edge on the original image which lay outside the 8x8 array had the value zero. The treatment of the degradation at the borders of the image does not affect the values of $\underline{A}$ considerably. However, the operation of the SVD method is greatly affected for some unknown reason. Evidently, some difficulty arises in the calculation of the singular values. The matrix $\underline{A}$ which was given earlier, though, causes no problems in the operation of the SVD method.

The two methods were compared for different amounts of zero-mean additive Gaussian noise. The degraded image plus noise that has a standard deviation of $\sigma = 0.1$ is shown in Fig. 3. The image restored by the SVD after 48 terms is given in Fig. 4a. The projection algorithm result after 15 iterations is shown in Fig. 4b. This restoration is about equivalent to the SVD. The projection algorithm can include a priori information. If the image vector $\underline{f}$

is restricted to positive values after each iteration, the restoration improves markedly (Fig. 4c,d). From Fig. 4a,b,c,d, it is evident that the projection algorithm with the positivity constraint yields a better restoration for this level of noise.

In Fig. 5 a,b the noise is increased to $\sigma = 0.5$. The best results from each of the two methods is presented. The choice between the two is difficult. The projection method image is "cleaner", but the SVD result is less "checkered" in appearance. For the latter reason, the SVD image is probably preferable.

The noise level in Fig. 6a,b is $\sigma = 1.0$. The two results are similar. But the SVD result is slightly better for the same reasons as stated for Fig. 5. In Fig. 7a,b, both methods are near the limit of performance at the noise level of $\sigma = 1.5$. The SVD image is more cluttered, but perhaps is more recognizable as a "5".

The "error" between the original and restored images was recorded along with the visual results. The error was calculated as: Error = $\sum_{i=1}^{n} (f_i - \hat{f}_i)^2$, where n = 64 for this case. It is doubtful whether this error is a useful quantity in the comparison of restored images. For example, in one instance of the projection method an increase of 13% occurred between the 25th and 70th iterations. However, the restored image improved slightly at iteration 70. So the image with the least error (as calculated above) is not necessarily the best.

The computer time required by the two methods was also observed. For this problem the projection algorithm required 0.16 seconds for preliminary calculations and 0.173 seconds for each iteration. The time required to generate the pictures was not included. The SVD took about 9.0 seconds to read the eigenvectors and singular values from magnetic tape. Only 0.08 seconds were needed to generate the 48th term. However, about 60 seconds were needed to generate the 48th term. However, about 60 secs. were needed to generate the eigenvectors and singular values and store them on magnetic tape. Once the eigenvectors are generated, they can be used in the SVD restoration of any image which was blurred by the same degradation. The major time requirement for the SVD is the retreival of the stored information.

The projection algorithm is seen to yield better results than the SVD for low noise levels. The SVD gives somewhat better restorations than the projection method as the additive noise is increased. Once the eigenvectors are stored, the SVD computer time is comparable to that required by the projection method. It is difficult to evaluate the computer efficiency of the SVD for this 8x8 pixel problem. As the size increases, the SVD implementation becomes much more difficult. The difficulty in the projection method, on the other hand, depends more on the extent of the degrading system impulse response.

V. Conclusions

The comparison between the singular value decomposition and the projection algorithm has yielded some mixed results. The projection algorithm has proven to be an effective method of restoration. For the particular example presented in this report, the projection method is definitely better than the SVD for low noise levels. However, the higher levels of additive noise tend to reverse this, and the SVD gives restored images whose appearance is subjectively better.

Even for this small example, the computer time required to generate the eigenvectors and singular values for the SVD is large. One of the advantages of the projection algorithm is that it can handle large images much more readily than the SVD. The projection method is also more versatile in that it can handle a priori information about the original image. Both methods rely on a subjective determination of the number of terms or iterations required to give the best restoration.

The next report will contain a one-dimensional problem for further comparison between these two methods.

References

[1] T. S. Huang and P. M. Narendra, "Image Restoration by Singular Value Decomposition," Applied Optics, Vol. 14, p. 2213, Sept. 1975.

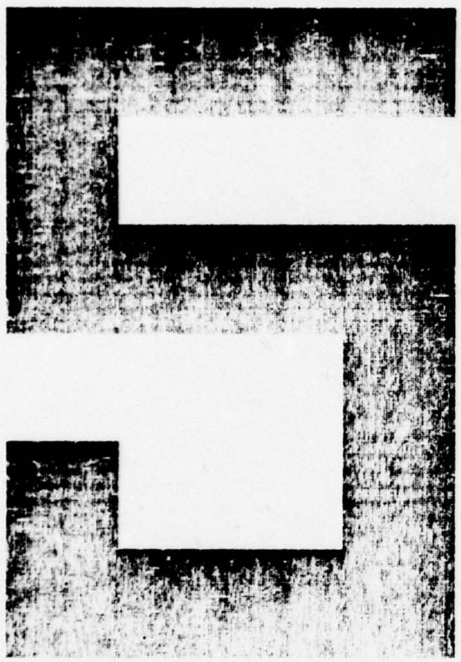


Figure 1 Original image

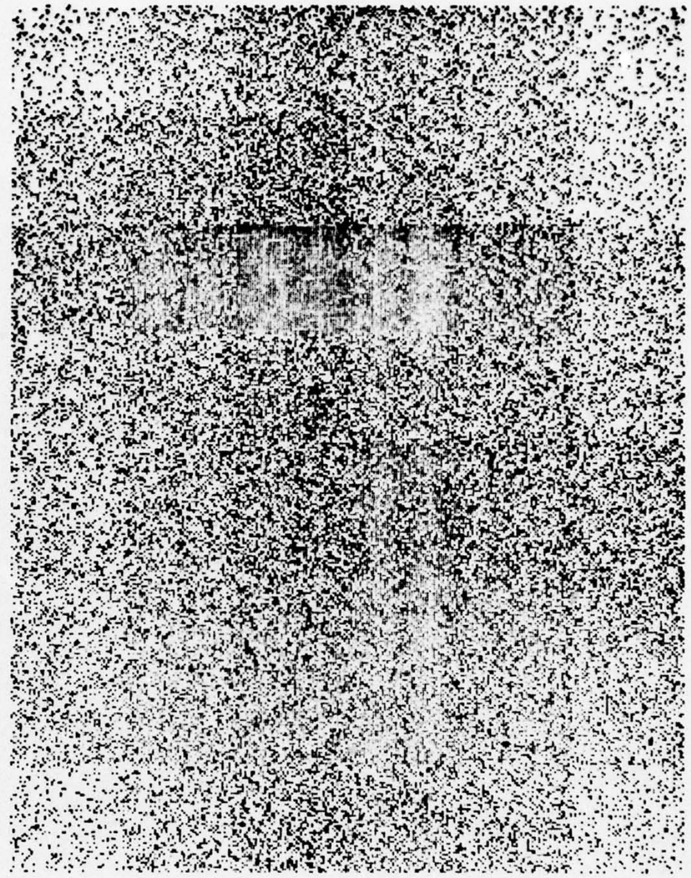


Figure 2 Linearly degraded image

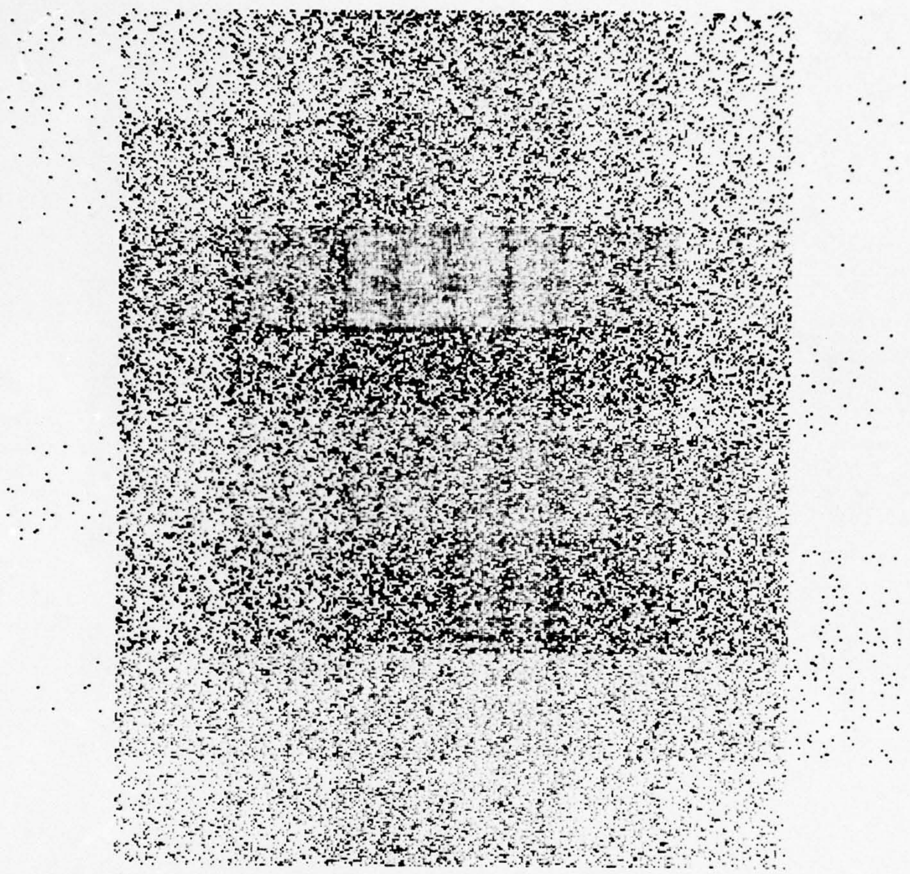


Figure 3 Degraded image plus noise, $\sigma = 0.1$

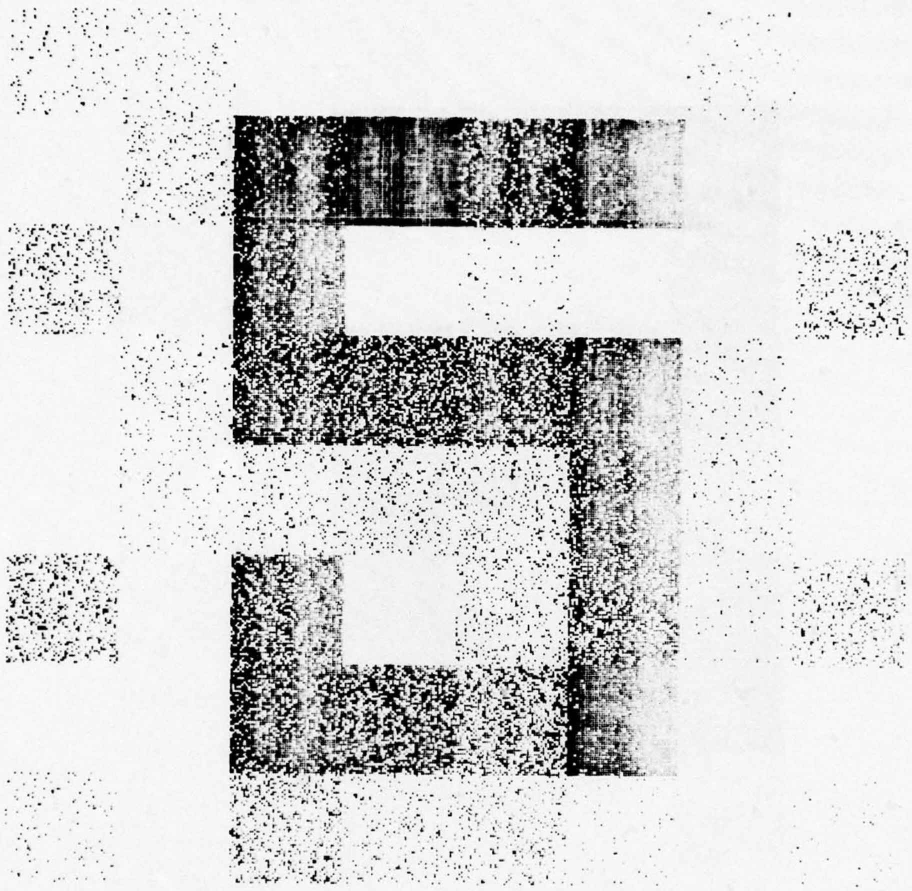


Figure 4a SVD restored image, 48 terms, $\sigma = 0.1$

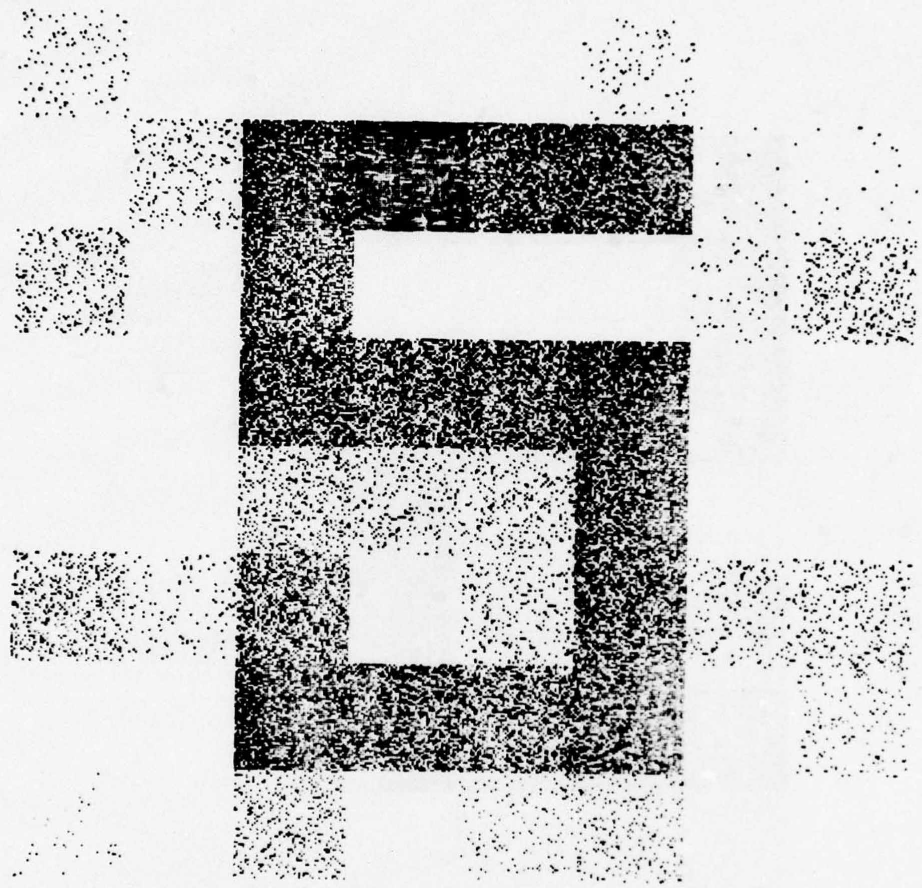


Figure 4b Porjection algorithm, 15 iterations, $\sigma = 0.1$

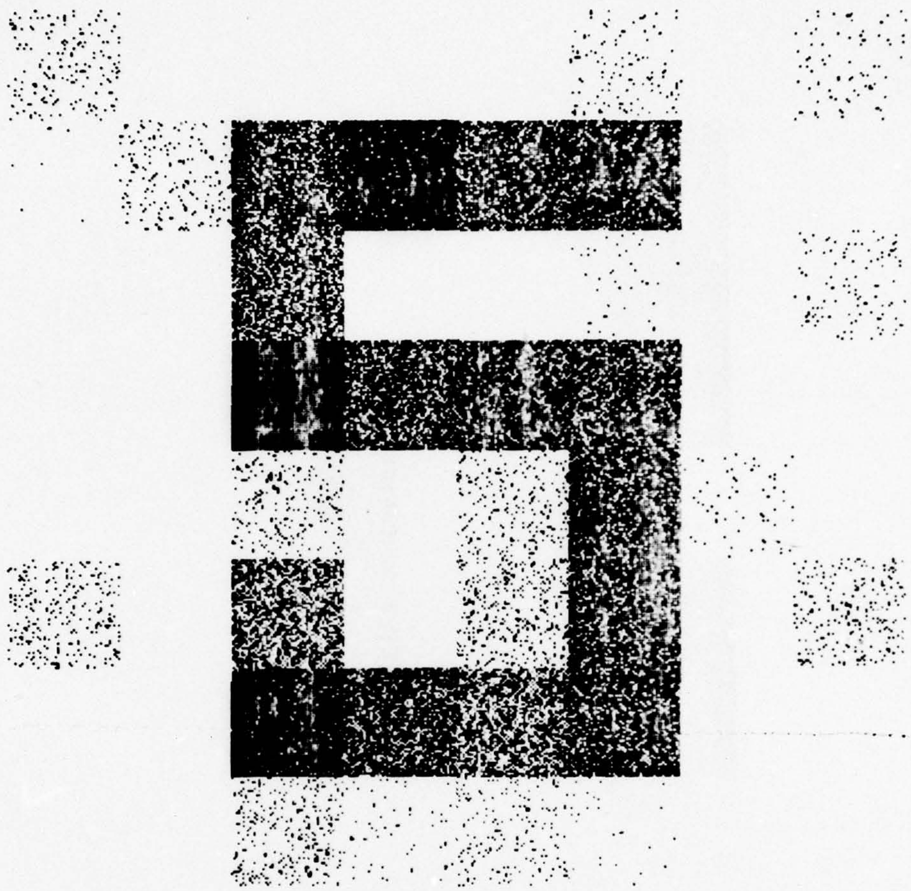


Figure 4c Projection algorithm, positivity constraint $-\underline{f} \geq 0$, 15 iterations, $\sigma = 0.1$

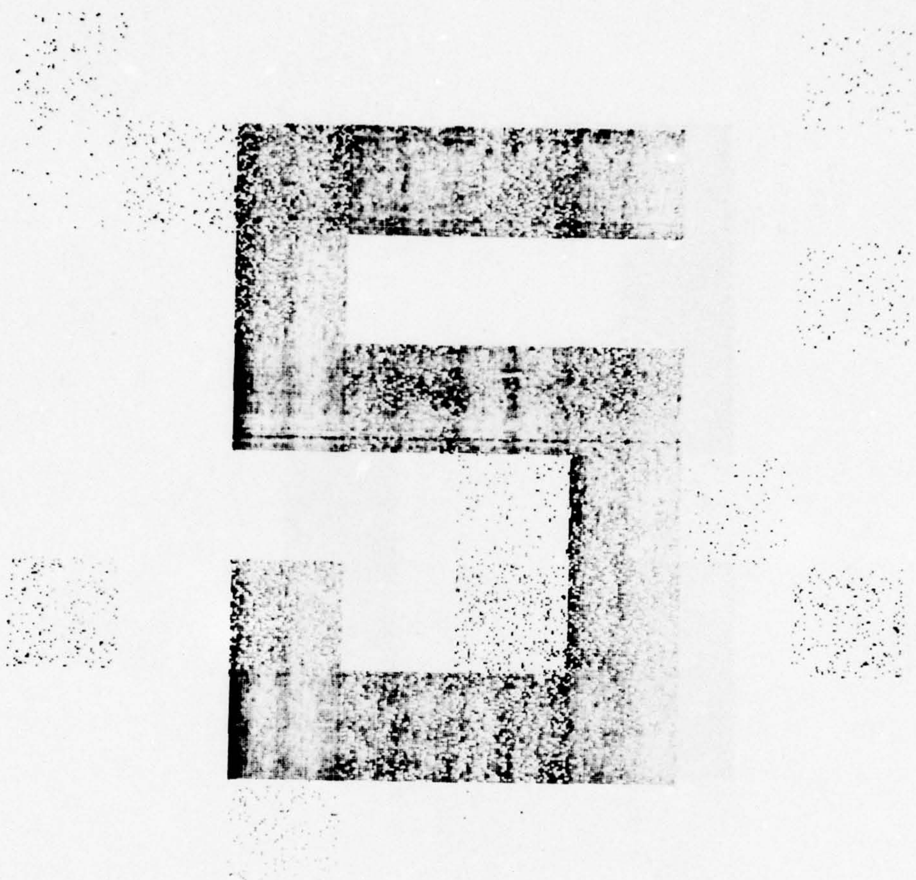


Figure 4d Projection algorithm, $f \geq 0$, 30 iterations, $\sigma = 0.1$

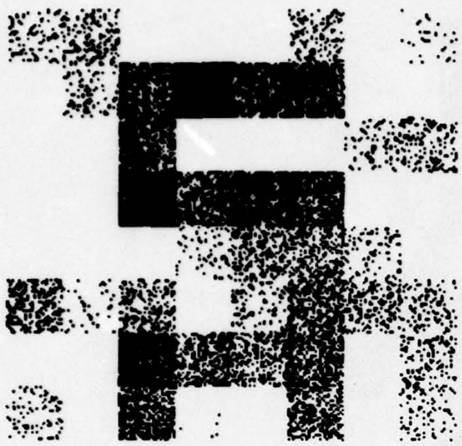


Figure 5a SVD, 44 terms, $\sigma = 0.5$

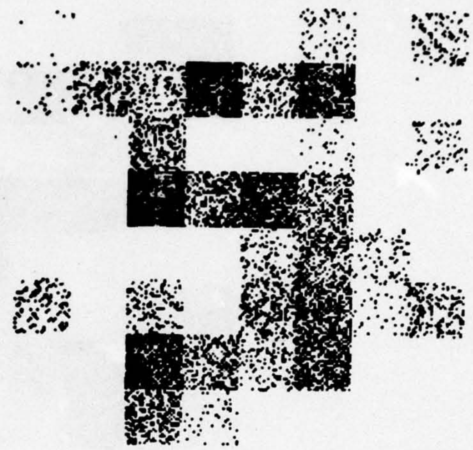


Figure 5b Projection algorithm $f \geq 0$, 8 iterations, $\sigma = 0.5$

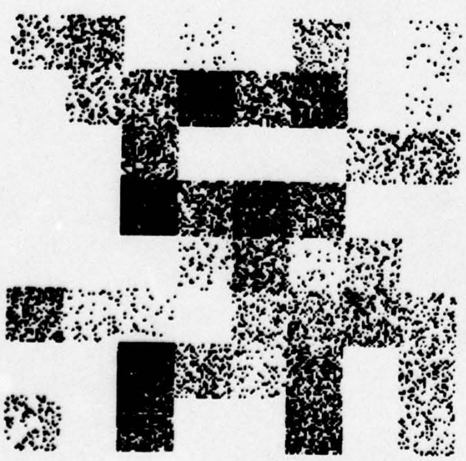


Figure 6a SVD, 44 terms, $\sigma = 1.0$

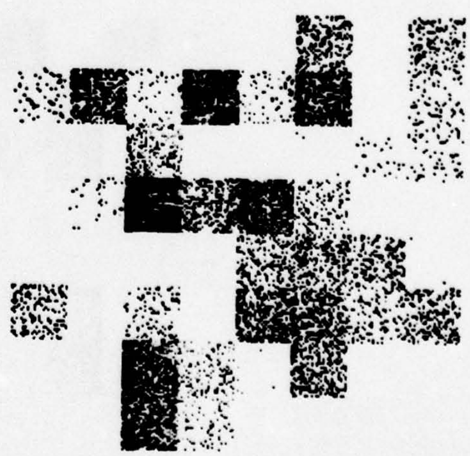


Figure 6b Projection algorithm $f \geq 0$, 4 iterations, $\sigma = 1.0$

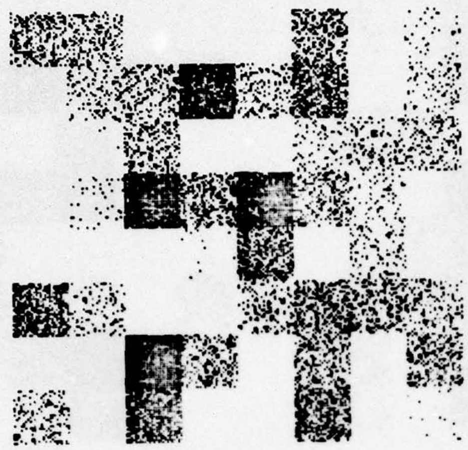


Figure 7a SVD, 40 terms $\sigma = 1.5$

Figure 7b Projection algorithm, $f \geq 0$, 6 iterations, $\sigma = 1.5$

FOURIER DESCRIPTORS AND THEIR APPLICATION TO AIRPLANE SHAPE ANALYSIS

T. Wallace and P. A. Wintz

In our previous two progress reports, we have discussed the practical implementation of classification of shapes using Fourier descriptors. The basic operations of scaling, rotation, and changing the starting point of a sampled contour have been discussed from the frequency domain viewpoint, folfowing the theoretical development of Granlund [1]. The major difficulty encountered was normalizing these factors in a way which preserved all of the shape information in the space domain representation. It was shown that the magnitude information is more important than the phase information for bilaterally symmetric contours such as the airplane outlines we have been working with recently. While simple magnitude normalization avoids the problems associated with defining a unique orientation and starting point, some information is necessarily lost.

Some normalization schemes which have been proposed work in the absence of noise, but give poor results when applied to real data. The simplest normalization scheme is to constrain frequency coefficients A(1) and A(2) to some reference phase, performing the two normalization operations to achieve this result. For some data sets, this method is less successful than that using the magnitude information alone, due to noise which perturbs the orientation and starting point resulting from normalization. We have now developed a method of FD normalization which preserves all of the shape information while rejecting noise successfully. In order to reject noise, the coefficients used in the procedure are chosen to have as large magnitudes as possible.

First, we require the phases of the two largest coefficients to be zero. A(1) will always be the largest, with magnitude unity due to the scale normalization procedure which defines that magnitude. Let the second largest coefficient be A(k). (The frequencies of the coefficients produced by an FFT of length n range from $-(n/2) + 1$ to $(n/2)$). The normalization multiplicity m of coefficient A(k) is defined as:

$$m = |m-1|$$

Theorem: The requirement that A(1) and A(k) have zero phase angle can be satisfied by m different orientation/starting point combinations.

PROOF: Use the two allowable operations to arrive at one orientation and starting point which gives zero phase for A(1) and A(k). Next use the starting point movement operation (multiplication of the ith coefficient by $\exp(I \cdot J \cdot T)$) to move the starting point once around the entire contour. To accomplish this I must range from 0 to 2π. Now consider the two cases k positive and k negative. If k is positive, the phase of A(1) and A(k) will coincide at k-1 different starting points. But at each of these starting points, we can use the orientation operation (multiplication of each coefficient by $\exp(j \cdot j\theta)$) to reduce the phases to zero. Similarly, if k is negative, the phases of A(1) and A(k) will coincide at 1-k different starting points. Again, the orientation operation can reduce the phases to zero.

Note that if k=2, the orientation and starting point are defined uniquely. In general, however, A(2) will not be the second largest coefficient in magnitude so this ambiguity must be resolved to achieve a general procedure.

The obvious method of solving this problem is to check the phase of a third coefficient A(p) at each of the m possible orientation/starting point combinations and choose the normalization which gives a phase closest to zero for this coefficient. However, this ambiguity-resolving coefficient cannot be chosen arbitrarily. If the normalization multiplicity of coefficient A(p) is the same as that of A(k), or a multiple of it, the phase of A(p) will be the same at each possible normalization! If m for coefficient A(p) (denoted m[p]) is a factor of m[k], or a multiple of a factor of m[k] less than m[k], there is also ambiguity since some of the m possible normalizations will result in identical phases for A(p). If these ambiguous coefficients are removed from consideration, and the unambiguous coefficient with the largest magnitude is used to select one of the m allowable normalizations, a general procedure is obtained.

To briefly review the entire normalization procedure, we start by dividing each coefficient by the magnitude of A(1) to normalize the size of the contour. We find the coefficient of second largest magnitude and compute its normalization multiplicity. We then locate the third largest coefficient suitable for resolving the ambiguity (A(p)) as explained above. The orientation and starting point are adjusted to satisfy the restrictions that A(1) and A(k) are real and positive, and A(p) has phase as close to zero as possible.

This method is quite powerful, but a slight modification in the procedure has been found helpful in those cases in which there are two or more coefficients suitable for use as A(p) with almost the same magnitude. It is very unlikely that the magnitudes will be identical, but if they are even close, noise may cause one of them to be used to normalize the test FD, and the other to normalize the unknown FD. To overcome this, the ambiguity-resolving coefficient used to normalize the test FD can be supplied to the normalization subroutine directly, rather than having the subroutine compute it.

To investigate the classification accuracy using this method as opposed to just using the magnitudes, the experiment described in our last progress report was performed both ways. Briefly, the experiment involved classifying 20 aircraft contours quantized to a 64x64 grid using a test set consisting of the same aircraft contours but quantized to a resolution of 128x128. Using this method, the classification accuracy was 100% for an absolute value distance measure, and 95% for a Euclidean distance measure, as reported previously. Using only the magnitude information, classification accuracy dropped to 95% and 90% respectively.

The next step in this research involves using the BLOB algorithm to locate the outlines of objects in actual photographic data, then classifying the shapes using the Fourier Descriptor algorithm described above. The version of BLOB that we are presently working with differs from earlier versions in that it does not preferentially find outlines in any given direction. There is hence very little dependency on the direction in which the picture was scanned, and no consistent false contours in any single direction.

The major problem in adapting BLOB to this application involves the statistical assumptions used to classify pixel groups as similar or dissimilar. The original BLOB was developed to classify multispectral data, which was assumed normally distributed. This assumption is generally ineffective in dealing with ordinary photographic data, and preliminary results reflect this fact. More appropriate statistical assumptions are under consideration in order to improve contour location accuracy.

REFERENCES

[1] G. H. Granlund, "Fourier Preprocessing for Hand Print Character Recognition," IEEE Trans. on Computers, Vol. C-21, pp. 195-201, Feb. 1972.

LOCATING AIRPORTS IN LANDSAT IMAGERY

X.K. Dang and T.S. Huang

As mentioned in our report (Nov. 75 - Jan. 76) our concern is to separate airports from highways. We specify our visual model of a runway as follows:

1. Spectral properties of runways and highways are very similar corresponding to concrete materials.

2. The shape of a runway is different from a highway. It is a strip with the following properties:

   a) Large width

   b) Maximal length

   c) No curvature

3. Runways are not isolated. They form special networks.

Using the syntactic approach we construct the separation algorithm based on the first two properties of the runway model: (1) The spectral properties of concrete which set the threshold for the possible candidate points of the runway - highway complex and the shape of a runway which allows us to separate it from a highway. We have tested our algorithm on a 128x128 picture of INDIANAPOLIS where we can see WEIR COOK airport surrounded by highways and freeways (Fig. 1). The result is displayed in Fig. 2.

Details of this algorithm will be described in the next report. We are pursuing our work to locate the runways using two techniques: (1) a spatial frequency filter to remove all non-linear objects in Fig. 2; (2) a higher level syntactic algorithm to remove all non-runway objects in Fig. 2.

Fig. 1 Indianapolis Airport

Fig. 2 After highway filtering

FLIR IMAGERY TACTICAL TARGET DETECTION AND CLASSIFICATION

O. R. Mitchell

This project is being carried out in cooperation with Honeywell's System and Research Division, Minneapolis, Minnesota. Honeywell has developed a real-time target cueing system, the Autoscreener, which is geared to the detection of tactical targets in a rural environment. Purdue has agreed to assist in developing improved techniques for this device to allow target detection in an urban environment and target classification in both rural and urban environments.

The present direction of our work is to develop algorithms for target classification in a rural environment. The lack of urban FLIR data containing tactical targets has led us to concentrate initially on the rural case. Our overall goal is to automatically rapidly classify objects into vehicles and non-vehicles and to further classify each category into several tactical classes (e.g. tank, APC, jeep, truck).

We plan to use statistical, textural, and shape information in segmenting a designated target into subparts. These subparts might include: motor hot spot, wheels or tracks, body, windshield, turret, etc. Classified subparts would then be checked against a grammatical description of desired targets.

A more global clue which might be helpful in a rural classification is a knowledge of the terrain type as detected by a texture measure. For example, the shape constraints for a target in a heavily wooded area should be relaxed somewhat since some parts of the object may be obscured by trees.

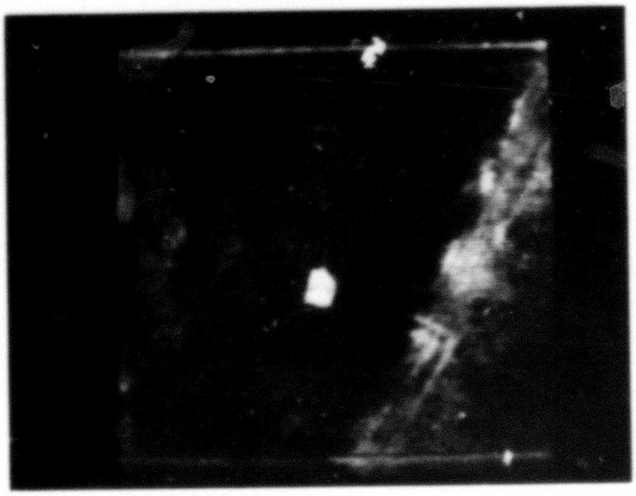
The inclusion of urban environment targets will depend heavily upon image segmentation and background classification. The shape constraints or target objects will allow partial target obstruction by other man-made objects.

Moving objects may be detected by using multiframe change detection.

These proposed techniques require use of existing segmentation, feature extraction, and syntactic algorithms; the development of new methods due to the unique character of FLIR data; and the implementation of this software efficiently in a real-time system.

A data set has been selected for initial work. This consists of 25 digitized 480×512 images, each containing one, two, or no targets. Sample targets are shown in Fig. 1. Some presently available algorithms have been applied to these images. For example, the texture edge algorithm (see report in prior section) produced the output shown in Figs. 2-4 and the statistical contour following algorithm produced the contour shown in Fig. 5.

(a)



(b)

Figure 1 Sample FLIR imagery. Target is in center.

Figure 2 Texture edges in Fig. 1(a) found by comparing local extrema surrounding each part.

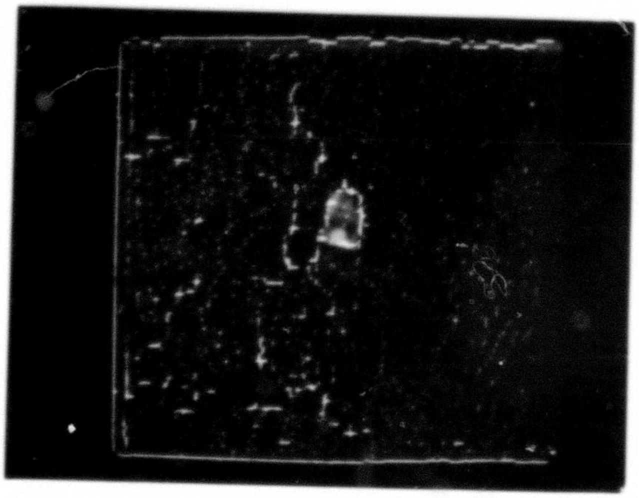


Figure 3 Local maxima in Fig. 2 superimposed on Fig. 1(a).

Figure 4 Each horizontal interval shown has the correct texture and intensity to be a target (original data in Fig. 1(b).

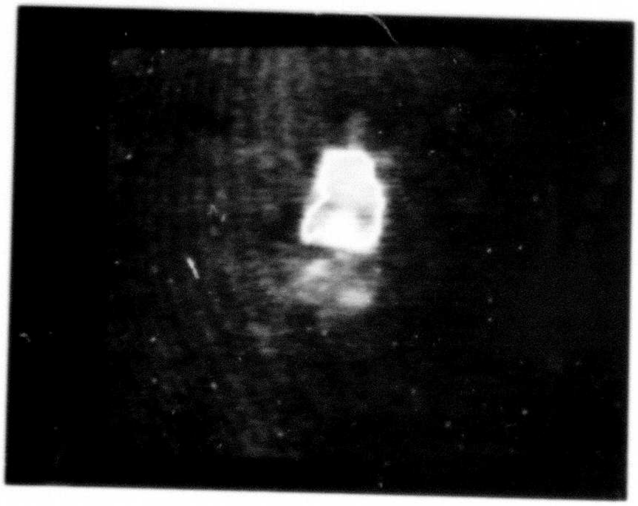


Figure 5 Output of statistical contour following algorithm.

FACILITIES

QTY	Manufacturer	Description
3	Beehive Elect.	"Super-Bee" Terminals
2	Tex. Inst.	"Silent 700" Terminals
1	Digi-Data	Industry standard magnetic tape system; 2, 9-track and 1, 7-track drives; one each NRZI and phase-encoded formatters/controllers
1	DEC	Dual-drive DECtape unit
1	DEC	RP03 disk drive (40 million characters)
1	Fabritek	96K-word auxiliary memory system (64K bought by ARPA, 32K by NASA)
1	Versatek	Electrostatic matrix printer
1	Comtal	Color picture display
1	Data Printer	132 column, 600 L.P.M. line printer
1	True-Data	Punched card reader
1	Tektronix	Model 4010, graphics display
1	DEC	PDP-11/45 computer system; system includes: 32K memory FPP-11 floating point processor (NSF money) H960 extension mounting cabinet 3 - small peripheral mountings blocks (DD-11) 1 UNIBUS repeater/expander DH11, 16-line terminal multiplexor KW11-p programmed clock "ANTS" - type PDP-11/IMP interface

Note: Our PDP-11/45 is currently operating under the UNIX system.

BOOKS

FU, K.S., "ERTS Data Analysis," chapter in Application of Syntactic Pattern Recognition, ed., Springer-Verlag, 1976 (with J. Brayer, P. H. Swain).

SWAIN, P.H., "ERTS Data Analysis," chapter in Applications of Syntactic Pattern Recognition, K. S. Fu, ed., accepted for publication by Springer-Verlag, 1976.

WINTZ, P.A., "Picture Coding and Feature Extraction," chapter in Digital Image Processing, 1976.

JOURNAL PUBLICATIONS

FU, K.S., "An Application of Stochastic Languages to Fingerprint Pattern Recognition," Pattern Recognition, Vol. 8, pp. 175-181, 1976 (with B. Moayer)

FU, K.S., "A Minicomputer Facility for Picture Processing and Pattern Recognition Research," COMPUTER, Vol. 9, pp. 70-77, May 1976 (with E. Persoon)

FU, K.S., "Parametric Feature Extraction Through Error Minimization, Applied to Medical Diagnosis," IEEE Trans. on Systems, Man, and Cybernetics, Vol. SMC-6, September 1976 (with T. Lissack)

FUKUNAGA, K., "A Graph-Theoretic Approach to Nonparametric Cluster Analysis," IEEE Trans. on Computers, Vol. C-25, pp. 936-944, Sept. 1976 (with W.L.G. Koontz, P.M. Narendra)

MITCHELL, O.R., "Digital Communications Equipment for Instructional Purposes," IEEE Trans. on Education, Vol. E-19, No. 2, pp. 66-70, May 1976 (with W. L. Thomas)

CONFERENCES

FU, K.S., "Processing of Chest X-Ray Images by Computer," IFIP Working Conference on Decision-Making and Medical Care, May 24-29, 1976, Dijon, France.

FU, K.S., "High Dimensional Languages and Grammatical Inference," IEEE Joint Workshop on Pattern Recognition and Artificial Intelligence, June 1-3, 1976, Hyannis, MA.

Fu, K.S., "Some Applications of Stochastic Languages," Symposium on Application of Statistics, June 14-18, 1976, Dayton, Ohio.

FU, K.S., "Tree System Approach for LANDSAT Data Interpretation," Proc. Symp. on Machine Processing of Remotely Sensed Data, June 29-July 1, 1976.

FU, K.S., "An Approach to the Design of a Linear Binary Tree Classifier," Proc. Symp. on Machine Processing of Remotely Sensed Data, June 29-July 1, 1976.

FU, K.S., "The Linguistic Approach to Pattern Recognition," Advanced Seminar on Classification and Clustering," The Mathematics Research Center, University of Wisconsin, Madison, Wisconsin, May 3-5, 1976.

HUANG, T.S., "Nonlinear Estimation of Markov Jump Processes," presented at the IEEE Int'l Information Theory Symposium, Ronneby, Sweden, June 21-24, 1976 (with J. Burnett)

HUANG, T.S., "Digital Straight Edges," presented at the 6th Annual Symp. on Automatic Imagery Pattern Recognition, Univ. of Maryland, Silver Spring, MD, June 1-2, 1976 (with G. Tang)

HUANG, T.S., "Two-Dimensional Fourier Transforms," "Image Restoration," and "Film Models," presented at NATO Advanced Institute on Digital Image Processing and Analysis, Bonas, France, June 14-25, 1976.

HUANG, T.S. and Mitchell, O.R., "Subjective Effect of Two-Dimensional Noise," SPSE, Symp. on Image Evaluation, July 19-23, 1976, Toronto, Canada.

HUANG, T.S., "Image Processing Research at Purdue," presented at Los Alamos Scientific Laboratory, Los Alamos, NM, May 19, 1976.

MITCHELL, O.R., "Texture Edge Detection and Classification Using Max-Min Descriptors," Sixth Annual Symposium on Automatic Imagery Pattern Recognition, Univ. of Maryland, College Park, MD, June 1-2, 1976.

MITCHELL, O.R., "Filtering to Remove Cloud Cover in Satellite Imagery," LARS Symposium, Machine Processing of Remotely Sensed Data, June 29-July 1, 1976, West Lafayette, IN (with P. L. Chen)

MITCHELL, O.R. and HUANG, T.S., "Subjective Effect of Two-Dimensional Noise," SPSE Symp. on Image Evaluation, July 19-23, 1976, Toronto, Canada.

SWAIN, P.H., "Some Time for Texture in the Spectrum of Spatial Features," presented at the Engineering Foundation Conference on Algorithms for Image Processing, Franklin Pierce College, Rindge, NH, August 1976 (with D. A. Landgrebe)

WINTZ, P.A., "Images and Models for Image Noise," presented at NATO Advanced Institute on Digital Image Processing and Analysis, Bonas, France, June 14-25, 1976.

WINTZ, P.A., "Image Coding with Emphasis on Techniques for Producing Decorrelated Image Data," presented at NATO Advanced Institute on Digital Image Processing and Analysis, Bonas, France, June 14-25, 1976.

STAFF

CO-PRINCIPAL INVESTIGATORS

T. S. Huang
K. S. Fu

PROFESSORIAL

K. Fukunaga
O. Mitchell
P. Swain
P. Wintz

GRADUATE RESEARCHERS

S. Berger
J. Burnett
S. Carlton
W. Chan
P.H. Chen
P.L. Chen
X. Dang
R. Florek
J. Keng
R.L. Li
Y.K. Lin
P. Narendra
B. O'Connor
D. Panda
W. Pfaff
A. Salahi
R. Short
G. Tang
T. Wallace
M. Yoo
T.S. Yu

RESEARCH STAFF

J. Besemer
W. Robey

UNDERGRADUATE RESEARCHERS

M. DeMoney
T. Hawker
J. Meese
J. Schwab
B. Zurney

ELECTRONIC TECHNICIANS

D. Azpell
P. Crane
J. Rogers
F. Woodworth

SECRETARIES

M. Barbour
M. Claire

METRIC SYSTEM

BASE UNITS:

Quantity	Unit	SI Symbol	Formula
length	metre	m	...
mass	kilogram	kg	...
time	second	s	...
electric current	ampere	A	...
thermodynamic temperature	kelvin	K	...
amount of substance	mole	mol	...
luminous intensity	candela	cd	...

SUPPLEMENTARY UNITS:

Quantity	Unit	SI Symbol	Formula
plane angle	radian	rad	...
solid angle	steradian	sr	...

DERIVED UNITS:

Quantity	Unit	SI Symbol	Formula
Acceleration	metre per second squared	...	m/s
activity (of a radioactive source)	disintegration per second	...	(disintegration)/s
angular acceleration	radian per second squared	...	rad/s
angular velocity	radian per second	...	rad/s
area	square metre	...	m
density	kilogram per cubic metre	...	kg/m
electric capacitance	farad	F	A·s/V
electrical conductance	siemens	S	A/V
electric field strength	volt per metre	...	V/m
electric inductance	henry	H	V·s/A
electric potential difference	volt	V	W/A
electric resistance	ohm		V/A
electromotive force	volt	V	W/A
energy	joule	J	N·m
entropy	joule per kelvin	...	J/K
force	newton	N	kg·m/s
frequency	hertz	Hz	(cycle)/s
illuminance	lux	lx	lm/m
luminance	candela per square metre	...	cd/m
luminous flux	lumen	lm	cd·sr
magnetic field strength	ampere per metre	...	A/m
magnetic flux	weber	Wb	V·s
magnetic flux density	tesla	T	Wb/m
magnetomotive force	ampere	A	...
power	watt	W	J/s
pressure	pascal	Pa	N/m
quantity of electricity	coulomb	C	A·s
quantity of heat	joule	J	N·m
radiant intensity	watt per steradian	...	W/sr
specific heat	joule per kilogram-kelvin	...	J/kg·K
stress	pascal	Pa	N/m
thermal conductivity	watt per metre-kelvin	...	W/m·K
velocity	metre per second	...	m/s
viscosity, dynamic	pascal-second	...	Pa·s
viscosity, kinematic	square metre per second	...	m/s
voltage	volt	V	W/A
volume	cubic metre	...	m
wavenumber	reciprocal metre	...	(wave)/m
work	joule	J	N·m

SI PREFIXES:

Multiplication Factors	Prefix	SI Symbol
1 000 000 000 000 = 10^{12}	tera	T
1 000 000 000 = 10^{9}	giga	G
1 000 000 = 10^{6}	mega	M
1 000 = 10^{3}	kilo	k
100 = 10^{2}	hecto*	h
10 = 10^{1}	deka*	da
0.1 = 10^{-1}	deci*	d
0.01 = 10^{-2}	centi*	c
0.001 = 10^{-3}	milli	m
0.000 001 = 10^{-6}	micro	μ
0.000 000 001 = 10^{-9}	nano	n
0.000 000 000 001 = 10^{-12}	pico	p
0.000 000 000 000 001 = 10^{-15}	femto	f
0.000 000 000 000 000 001 = 10^{-18}	atto	a

* To be avoided where possible.

MISSION
of
Rome Air Development Center

RADC plans and conducts research, exploratory and advanced development programs in command, control, and communications (C^3) activities, and in the C^3 areas of information sciences and intelligence. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.

AMERICAN REVOLUTION BICENTENNIAL 1776·1976

☆U.S. GOVERNMENT PRINTING OFFICE: 1977-714-025/120